NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

IDENTIFYING UNAUTHORIZED DEVICES ON VLANs USING SOFTWARE-DEFINED NETWORKS

by

Vincent T. Amos

March 2019

Thesis Advisor:	Geoffrey G. Xie
Second Reader:	John D. Fulp

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2019	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE IDENTIFYING UNAUTHORIZED DEVICES ON VLANS USING SOFTWARE-DEFINED NETWORKS	**5. FUNDING NUMBERS**
6. AUTHOR(S) Vincent T. Amos	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Current naval networks vary in size and implementation, but have one thing in common: poor network device connectivity oversight. Poor network oversight can lead to unauthorized network access, but there is a potential solution with software-defined networking (SDN). SDN technology provides the management oversight and capability to maintain a complete network picture of all connected devices. SDN is the network technology that separates the control plane from the forwarding plane of the network while providing ability to program the entire network from a central controller. This thesis reviews the current network access control solution deployed for the NPS unclassified network and creates a SDN solution aimed to provide improvements in the following areas: a centralized network topology, low management overhead, and reduction in hardware and operational costs.

14. SUBJECT TERMS SDN, VLAN, 802.1x authentication, RADIUS	**15. NUMBER OF PAGES** 95
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

IDENTIFYING UNAUTHORIZED DEVICES ON VLANs USING SOFTWARE-DEFINED NETWORKS

Vincent T. Amos
Lieutenant, United States Navy
BS, University of Houston, 2006

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2019

Approved by: Geoffrey G. Xie
Advisor

John D. Fulp
Second Reader

Peter J. Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Current naval networks vary in size and implementation, but have one thing in common: poor network device connectivity oversight. Poor network oversight can lead to unauthorized network access, but there is a potential solution with software-defined networking (SDN). SDN technology provides the management oversight and capability to maintain a complete network picture of all connected devices. SDN is the network technology that separates the control plane from the forwarding plane of the network while providing ability to program the entire network from a central controller. This thesis reviews the current network access control solution deployed for the NPS unclassified network and creates a SDN solution aimed to provide improvements in the following areas: a centralized network topology, low management overhead, and reduction in hardware and operational costs.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Acronyms and Abbreviations

AAA	authorization, authentication and accounting
AMF	access management function
AD	active directory
API	application programming interface
AuthFlow	authentication flow
Bpps	billion packets per second
CA	certificate authority
CLI	command line interface
CAM	content-addressable memory
CORD	Central Offices Re-architected as Datacenters
DC	domain controller
DHCP	dynamic host configuration protocol
DNS	domain name server
DoD	Department of Defense
EAP	extensible authentication protocol
FlowNAC	flow-based network access control
GPT	GUID partition table
GUI	graphic user interface
GID	group ID

IEEE	Institute of Electrical and Electronics Engineers
IP	Internet Protocol
IT	information technology
ITACS	Information Technology and Communications Services
LAN	local area network
LDAP	lightweight directory access protocol
MAC	media access control
MASINT	measurement and signature intelligence
MBR	master boot record
NAC	network access control
NAS	network access server
NIC	network interface card
NOC	network operations center
NPS	network policy server
ODL	OpenDayLight
ONOS	Open Network Operating System
OS	operating system
PEAP	protected extensible authentication protocol
PPP	point-to-point protocol
RADIUS	remote authentication dial-in user service
RAM	random-access memory
SDN	software-defined networking

SNMP	simple network management protocol
SRWBR	short range wide band radio
Tbps	terabits per second
TCAM	ternary content-addressable memory
TCP	Transmission Control Protocol
USN	U.S. Navy
UDP	user datagram protocol
USG	United States Government
VLAN	virtual local area network
VM	virtual machine
WAN	wide area network

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgments

I would to take this time to thank my advisor, Dr. Geoffrey Xie, and second reader, J.D. Fulp, for being so patient throughout this process. Thank you, Mike Williams, for providing me with the hardware to pull off the ITACS implementation and for always being available for me to pick your brain about the ITACS network. And last but definitely not least, I would like to thank my wife, Maleah Amos, for being so patient and understanding when I had to stay at school late into the evening working to get this done. I think you pushed me more than my advisor did. And for that, I am forever grateful.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1: Introduction

This thesis evaluates how integrating software-defined networking (SDN) with virtual local area network (VLAN) technologies can improve the performance of identifying unauthorized devices in an enterprise network and, therefore, improve network security.

Naval ships have large networks with many hosts and devices. Spread throughout the ship, these hosts and devices are hard to keep track of once connected to the network. Often, hosts must be added or moved due to operational needs. The network administrators then go into the switches and authentication servers to add these devices manually, including adding their physical network interface addresses and port information to authorize them on the network. Manual configuration not only takes time, but also is error prone (e.g., attaching an unauthorized host to a restricted switch port). Fixing such errors after the fact further delays operability and thus the mission. Automating the port assignment process by implementing some type of authentication process with a network access server (NAS) is a solution widely deployed in enterprise networks.

This thesis reviews Naval Postgraduate School's unclassified network run by Information Technology and Communications Services (ITACS). Their network has an automated authentication process that assigns network access based off of one's username and password, along with whatever network policies for the organization and/or groups the account belongs to in the school's active directory (AD).

1.1 Problem Statement

The NPS network uses the Institute of Electrical and Electronics Engineers (IEEE) 802.1x authentication method to automatically assign newly attached machines to particular VLANs based on their physical network interface address. Although this implementation works, other network technologies may provide a more efficient way of authentication and network access control. SDN is a newer network technology that provides high programmability, which allows for dynamic, agile, and flexible networks, because SDN separates the forwarding functions from the network control plane. SDN allows for capabilities, such as network access control, to be programmed from a central location for the entire network, allowing a higher level of efficiency. However, several important questions must be answered to realize this potential. How do we go about building this type of network? How do we evaluate an ITACS SDN implementation to see how much more efficient it would be than the current ITACS implementation?

1.2 Research Questions

The current ITACS network has not been studied for a SDN solution. This thesis researches the following questions.

1. Does the SDN solution perform better than the current legacy VLAN solution when it comes to management overhead and knowledge of topology?
2. What are the pros and cons of this integrated approach?

1.3 Thesis Organization

This thesis is organized as follows. In Chapter 2, we define network access control. We break down different protocols and services used in implementing network access control on ITACS's network. We then define SDN and its primary parts and discuss related works. Chapter 3 discusses the high-level design of the replication of ITACS network and the design of the prototype SDN solution. In Chapter 4, we discuss our methodology, including how we built and configured the ITACS solution and the SDN solution, along with experiment results. Chapter 5 contains the conclusions drawn from our experimentation and discusses potential future work.

CHAPTER 2:
Background

In order to investigate the advisability of implementing SDN to improve network oversight and to reduce vulnerabilities as well as manual configuration, this chapter discusses the basics of network access control (NAC), including the ways it can be implemented and currently used protocols and services for doing so. These protocols and services are media access control (MAC) authentication, 802.1x, remote authentication dial-in user service (RADIUS), and lightweight directory access protocol (LDAP). We also review SDN and the different parts of its architecture.

2.1 Network Access Control

NAC is a network solution that uses a group of protocols and services to create policies that describe what credentials a device needs to join a network for the first time [4]. Some credentials that a device may need include username and password, required anti-virus update level, system specific update requirements, or device/MAC address authentication for printers, servers, or scanners. When users/devices do not meet these requirements, the NAC solution should deny access to network resources. One question, however, is what happens when legitimate users need to perform system updates in order to meet a NAC requirements? This is why remediation strategies exist.

Two of the remediation strategies are quarantine and captive portals [5]. Some NAC solutions quarantine the device using network segmentation and designating a certain network segment to deny access to the internet and all network resources except those that allow them, the user or device, to get up-to-date on all requirements to meet minimum policy standards. If a NAC solution uses the captive portal, it restricts users from accessing anything on the network and redirects all http/https request to the portal, usually an internal website, that provides them with the resources to download all updates and software requirements for them to become policy compliant and be granted access to the network. Some of the companies that provide what are considered the top NAC solutions (Extreme Works (Extreme Control), Auconet (BICS), Hewlett Packard Enterprise (ClearPass), ForeScout (CounterAct)) [6] use 802.1x authentication, network segmentation, MAC authentication, LDAP and an authentication server with various remediation techniques to provide their solutions [6].

2.2 Types of Access Controls

For this thesis, we consider network segmentation (VLAN), MAC authentication, 802.1x authentication, RADIUS, and LDAP as possible NAC solutions. We take a deeper look into these NAC methods.

2.2.1 VLAN

One solution to segmenting networks is to use VLANs. Rather than being separated by geographic location via physically separate devices (e.g., a network switch), VLANs are logically separated subnetworks of organizational groups, hosts/users, or devices that perform the same job functions. VLANs create separate broadcast domains [7]. Broadcast domains are typically bounded by routers because routers do not forward broadcast frames [8]. VLANs are the current standard for segmenting networks. VLANs are uniquely identified by a VLAN identifier (ID). Depending on the switch, the VLAN ID can be any number between 0-4094 [8]. The role of the VLAN ID is for it to be tagged in layer 2 Ethernet packets for routing purposes. If planned correctly, VLANs provide the desired logical groupings to perform access control for users and hosts to different applications and files within a wide area network (WAN) or local area network (LAN). Some other protocols and services can be leveraged to make VLANs more efficient. One of these services is MAC authentication.

2.2.2 MAC Authentication

A MAC address is a 48-bit address that is associated with a network interface card (NIC) to facilitate communication at the data link layer of the protocol stack. MAC addresses are commonly used in Ethernet and WiFi networks as the data link layer addresses [9]. MAC authentication is an authentication service that uses the MAC address of devices to gain–or not–access to network services. It is not the most secure, but it does provide a layer of security. This should not be the only layer of security unless perhaps one running a small, low risk, home network with a limited number of devices. If one is running a larger network, implementing other security protocols should be considered. MAC spoofing is an issue. Essentially, MAC spoofing entails changing a computer's MAC address, for any reason, and it is relatively easy to accomplish [10]. MAC spoofing can be accomplished by using specific tools that make an operating system (OS) believe its NIC has the MAC address of the user's choosing. Implementing the 802.1x protocol can help with this issue.

2.2.3 802.1x

The IEEE standard 802.1x is a standard for passing extensible authentication protocol (EAP) messages over a wired or wireless LAN [11].

> With a standardized EAP, interoperability and compatibility of authentication methods becomes simpler. For example, when you dial a remote-access server and use EAP as part of your PPP connection, the RAS doesn't need to know any of the details about your authentication system. Only you and the authentication server have to be coordinated. By supporting EAP authentication, a RAS server gets out of the business of acting as the middle man, and packages and repackages EAP packets to hand off to a RADIUS server that does the actual authentication. [11]

In ITACS's case, EAP is used with point-to-point protocol (PPP) connections. 802.1x authentication method works between three main device roles: the supplicant (the user/client), the authenticator, and the authentication server (RADIUS). [11] These three roles usually execute the protocols in the following order:

1. The supplicant connects to a port of an Ethernet switch.
2. The authenticator (switch) requests a username and password.
3. The authenticator sends the credentials to the authentication server (RADIUS).
4. The authentication server checks the credentials with known credentials in its database and sends back access granted/denied with required access policies to the authenticator.
5. The authenticator either allows/denies the user based on the reply from the authentication server.

The authentication server, RADIUS, needs further explaining.

2.2.4 RADIUS

RADIUS is a protocol built to support the client/server model. The client is typically a NAS or switch and the server is usually a UNIX/Windows based machine. It can also be described as a software package/protocol that provides authentication, authorization, and accounting services [12]. The client receives connection requests from users and sends their information to the RADIUS server. The RADIUS server checks the user's credentials, username and password, and then sends back the configuration information to the client stating what type of services to give to the user [13], [9]. RADIUS servers are not databases, though they do utilize databases, such as AD and MySQL, to provide authentication and authorization services. ITACS uses a Windows Server to provide RADIUS services. The RADIUS server is configured to use the Lightweight Directory Access Protocol (LDAP) to verify user credentials stored in the AD.

2.2.5 LDAP

LDAP is a software protocol that provides the ability to locate users, organizations, clients/devices and other resources on the network [14]. The protocol is used to retrieve and store information on a directory server. The LDAP protocol does not control how programs function on a client of the server, but transports the messages between the client and the directory server. LDAP rides over TCP, which means that a TCP session has to be established for this protocol to work [15]. When trying to authenticate between the server and client, the bind operation is used. The bind operation allow authentication information, sent by LDAP messages, to be exchanged between the server and the client. The bind operation is broken into three different messages: BindRequest, BindResponse, and UnbindRequest [16]. A BindRequest message is the initial message sent from the client to the server. The fields in message are the version, the name, and authentication. The version field contains the version of LDAP that is being used by the client. There is no negotiation of this field. If the server does not support the version of LDAP being used by the client, then the BindResponse from the server will state "protocolError" [16]. The name field contains the name of the directory object that the client is trying to bind as. It can be a null value if the client is trying to authenticate anonymously. The authentication field contains the information used to authenticate like username and password. The BindResponse message sends back the status of the client's request to authenticate. If the credentials are accepted by the server, then the ResultCode in the BindResponse will be set to "success" [16]. The

UnbindRequest is a message sent from the client to server to complete the unbind operation. The unbind operation is telling the server to quit or disconnect the LDAP session.

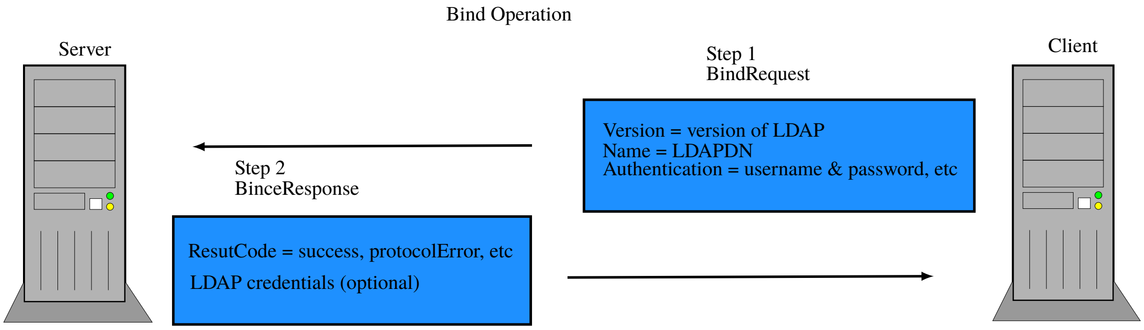

Bind Operation
Server
Client
Step 1
BindRequest
Version = version of LDAP
Name = LDAPDN
Authentication = username & password, etc
Step 2
BinceResponse
ResutCode = success, protocolError, etc
LDAP credentials (optional)

Figure 2.1. The Bind Operation Sequence

2.3 Software Defined Networks

One of the characteristics of SDN is separating the control plane from the forwarding plane. The forwarding plane contains all the logic and tables to deal with forwarding incoming packets based on networking characteristics, such as MAC addresses, Internet Protocol (IP) addresses and VLAN IDs [17]. When packets come to the forwarding plane, there are four ways they can be processed: forward, drop, consume, or replicate. The most basic functionality would entail a packet coming into the forwarding plane and being forwarded out the correct port after the address of the packet has been looked up in the forwarding table. If the packets address is not a part of the current forwarding table, it is consumed by the forwarding plane and passed on to the control plane for processing. The new forwarding table created by the control plane must be replicated to the forwarding plane before the packet can be sent to the proper port on the proper forwarding device. This is usually accomplished via a multicast sent out to all devices on the forwarding plane. The control plane determines the logic and algorithms being used and how they are configured at the forwarding plane. Prior to the development of SDN, the control and forwarding plane functions were always performed on the same device. By separating the two, the centralized control plane keeps all of the devices at the forwarding plane in sync, which theoretically eliminates loops [17]. From the concept of creating a separate control plane and forwarding plane, came the idea of creating a centralized controller. The concept was to remove the thousands of lines of control plane code [17] from the forwarding devices, thus making them more lightweight and agile to perform forwarding duties more quickly.

Conceptually, a SDN is made up of three main parts: SDN devices (switches), a controller, and applications. SDN devices are at the forwarding plane and contain forwarding functionality for packets. The data that is at each SDN device instructs the device where to send the packets. The data is represented by flows that are sent from the controller to the SDN devices [17]. A flow can be described as a list of instructions enumerating what to do with a group of packets from one end-point to another end-point or multiple end-points. The endpoints can be defined as IP addresses with TCP/UDP port pair, VLAN IDs and so on [17]. Flows are unidirectional, meaning that there has to be a flow for each direction even if it is between the same end-points. A list of flows are contained in a flow table on each SDN device. If a SDN device receives a packet that does not match a flow in its flow table, it sends the flow to the controller for instructions. The SDN controller is responsible for maintaining a current picture of all the SDN devices on its network. The controller also is responsible for sending this information to its northbound applications. The protocol used for communication between the SDN controller and the devices is called OpenFLow.

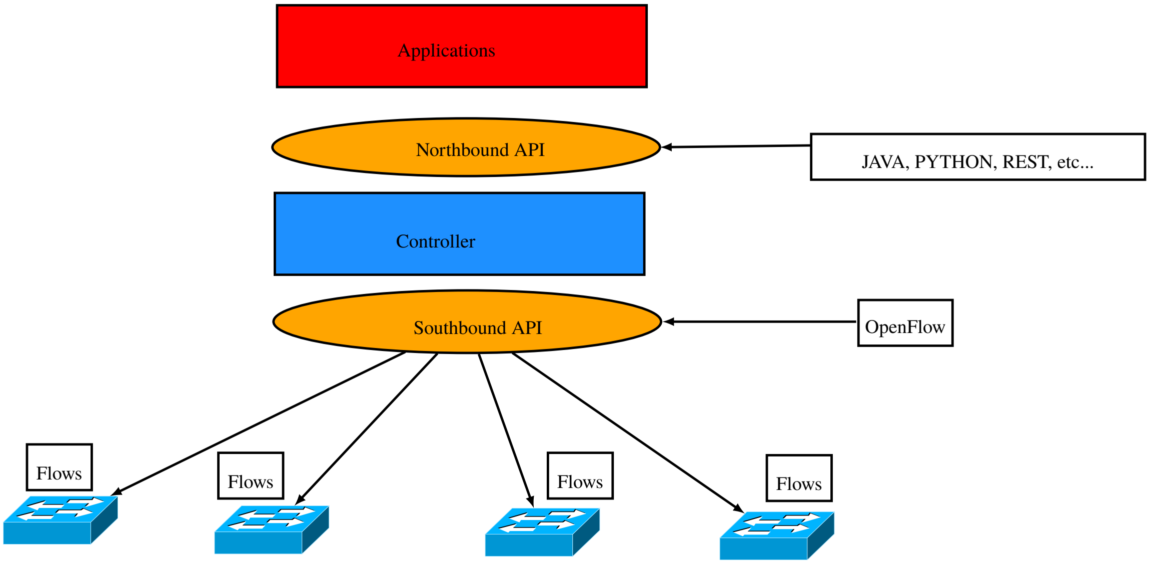


Figure 2.2. SDN Operations

2.3.1 OpenFlow

OpenFlow is the open source protocol for SDNs that allows the controller to add, update, or remove packet flows from the flow tables of SDN devices. OpenFlow runs over Transmission Control Protocol (TCP). The controller should be set up to listen on port 6653 for SDN switches that want to connect [18].

2.3.2 SDN Switches

There are two implementations of SDN devices: software and hardware. The software-based SDN devices are considered the preferred and easiest of the devices to implement, because flow tables are easily mapped by using either arrays, hash tables, or dictionary formats for flows. They are also dynamic in flow table sizing unlike the hardware SDN devices. The dynamic table growth allowed by software SDN devices means tables can grow in magnitudes larger than those of hardware SDN devices do to hardware technology constraints [17]. There are drawbacks though. Because of inefficient programming of search algorithms, searching through flow tables for software devices may lag when connected to really fast networks where link speeds can be 1Gbps or higher. Hardware SDN devices are a lot more efficiently built for quick reference to flow tables [17], [19]. This is why software SDN devices are typically recommended for small- to medium-sized networks, as large networks are in greater need of quicker flow table references made possible by hardware-based SDN devices.

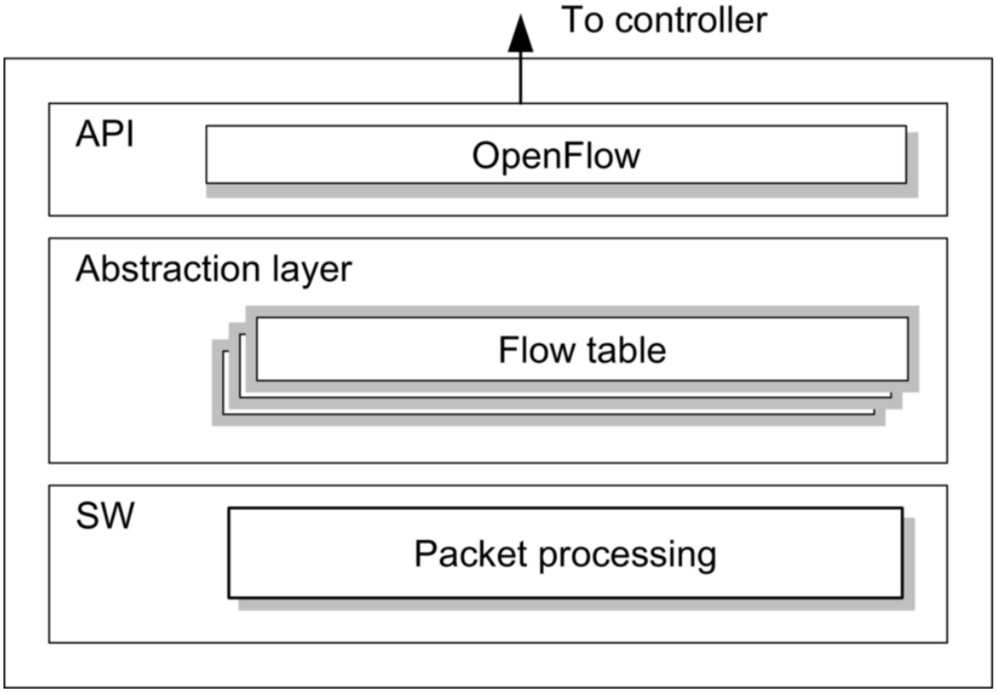


Figure 2.3. Software SDN Switch Architecture. Source: [17].

Hardware SDN devices replace packet matching software with specialized hardware. Hardware SDN devices work a lot faster than software SDN devices, which make them better for performance-sensitive networks such as datacenters and network cores [17]. There are two types of memory in which flow tables can be stored. Those two types of memory are content-addressable memory (CAM) and ternary content-addressable memory (TCAM). CAM is used for precise entries such as MAC addresses, which makes CAM good for layer 2 forwarding tables that is done by layer 2 switches [19]. TCAM is used for more complex forwarding functions. It accounts for a third state that is considered a wild card. For example, if the forwarding table was using IP addresses as its key, there might be multiple entries that match that IP address. That is where the wild card comes in. If wild card is the subnet mask, then it narrows down the list of flows in the flow table to send the packets to [17]. TCAM is used for layer 3 switches and routers for this reason [19]. Conventional switches and routers use random-access memory (RAM) for their flow table look ups because of the cost-to-speed benefits. CAM and TCAM is faster than RAM [19].

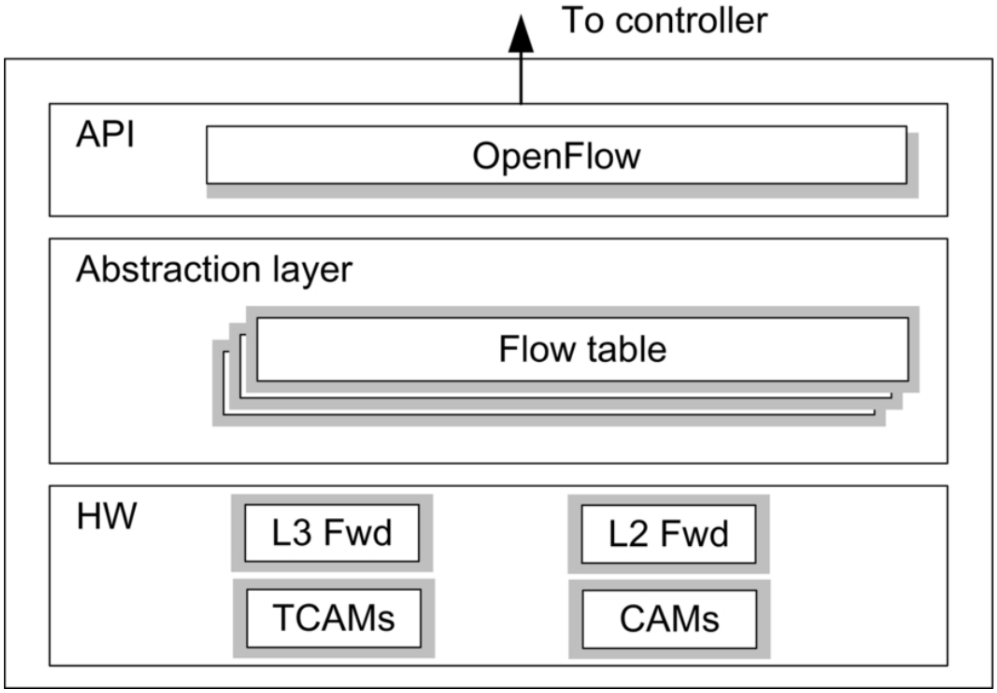


Figure 2.4. Hardware SDN Switch Architecture. Source: [17].

2.3.3 Controller

As previously stated, the controller maintains a current view of the entire network along with controlling policy updates concerning updating packet flow tables, routing, forwarding, redirecting packets, removing unreachable packet flows and load balancing [17]. Controllers do all these things in coordination with SDN applications. Controllers are able to communicate with these applications through application programming interface (API). These APIs form the northbound interface. Depending on the type of the controller being used, the northbound APIs could be implemented in JAVA, PYTHON, REST or many other languages. Northbound APIs can be low-level interfaces that only account for SDN devices in a common and more consistent manner [17]. This means that the API accounts for each SDN device. There are also northbound APIs with high-level interfaces that do not account for SDN devices at all, but only look at the overall network. Currently there is not a standard for northbound APIs for SDN controllers [17]. Many vendors are producing their own versions of SDN controller software. Some of those are OpenDayLight (ODL), Open Network Operating System (ONOS), Ryu, Cisco, VMWARE, Project Floodlight and

NOX [17], [20].

2.4 Related Works

802.1x port-based authentication provides a layer of reliable security for wired and wireless networks. Johan Loos wrote a paper on a wired implementation that walks you through his setup of 802.1x using Windows Server 2008 and 2012 [21]. He talks about the configurations of the domain controller (DC), AD, domain name server (DNS), certificate authority (CA) and dynamic host configuration protocol (DHCP) on Windows Server 2008 and network policy server (NPS) on Windows Server 2012 as a RADIUS server. He also describes the authentication process in great detail and with the authentication protocols: protected extensible authentication protocol (PEAP), acEAP-MSCHAPv2, and PEAP-EAP-MSCHAPv2 [21]. PEAP creates a secure tunnel connection between the authentication server and the client and then a EAP negotiation takes place between the client and the server. That is the major difference between the two. EAP-MSCHAPv2 authentication uses a password to authenticate and verified certificate chain. Combining these two authentication methods creates a secure tunnel established between the authentication server and client before passing the password for authentication. This combination of symmetric (password) and asymmetric (public and private keys) results in a protocol that is more secure than the two being used separately.

Building off of the 802.1x authentication wired implementation, our goal was to build a SDN 802.1x implementation that is competitive with, or outperforms it. Kamal Benzekki et al. took this concept and created a hybrid implementation where they removed the authentication process from the control plane and put it on the data plane to see if it would improve performance on a 802.1x authentication SDN network [22]. They studied different SDN and SDN-like networks to come up with their concept, such as:

- flow-based network access control (FlowNAC), which authenticates on the basis of flow nature [23].
- authentication flow (AuthFlow) which is an authentication and access control mechanism that is based on IEEE 802.1X and uses credentials in order to verify a host identity and determine the privileges using a POX controller, an authenticator and an authentication server [24].

- Flex access management system, which is based on OpenFlow and uses virtual network group ID (GID)s instead of VLANs for its access management mechanism. The GIDs are managed by access management function (AMF) that is a function on the OpenFlow controller. Clients are authenticate by MAC address via RADIUS [25].
- Access Control System for Wireless LAN roaming, the authors suggest new information for service provider's policy for the eduroam system that relies on 802.1x and a hierarchy of RADIUS proxies [26].

From these studies, they created to two implementations that they could test and analyze efficiency based on authentication latency and compare results. One was a conventional SDN network as shown in Figure 2.5 where the DHCP, RADIUS and AD servers are connected to the same OpenFlow switch as the controller. The other implementation has the DHCP, AD and RADIUS servers connected to a conventional switch that is connected to an OpenFlow switch that is connected to the controller as shown in Figure 2.6.

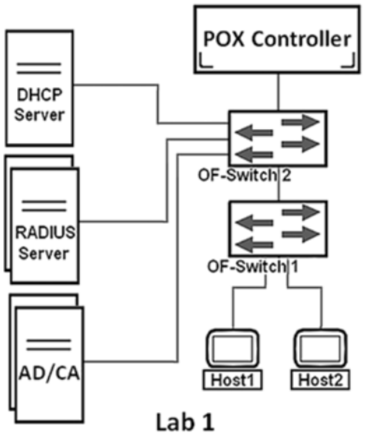


Figure 2.5. Conventional SDN Implementation. Source: [22].

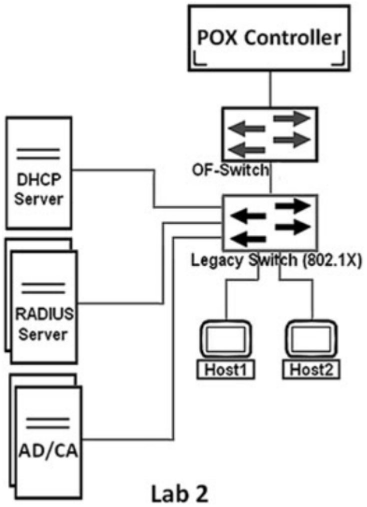


Figure 2.6. Conventional Switch Implementation. Source: [22].

The results from their study show that even when you put the authentication process on the data plane, in this case a conventional switch, it did not speed up the authentication process on SDNs. Placing your authentication process as close to the SDN controller as possible does; however, speed up that process.

CHAPTER 3: Design of Experiments

In this chapter, we discuss why we decided to create a replication of ITACS current network solution. We also discuss the design considerations for the SDN solution, which solution we chose, and why we decided to go with that solution.

3.1 Design Considerations

The goal of this thesis was to evaluate how SDN can improve network access control for ITACS. In order to evaluate this, the thesis built two types of networks in order to gain a better understanding of what configurations are necessary to implement and deploy all the devices and services needed to run their existing 802.1x authentication method. That information provides important design considerations for building our SDN prototype network for testing. SDN uses a different data communication protocol, OpenFlow, than the current ITACS solution. OpenFlow is the communication protocol that allows the SDN controller to interact with the forwarding plane network devices such as switches and routers [18]. In an ideal experimental environment, we would test a full replication of the current ITACS network and an SDN network at a similar scale to provide the most accurate data to ITACS. Due to research constraints, there were some design considerations and abbreviations of both networks that are in the following sections.

3.2 Replication of the ITACS Solution

Before we make any design choices, we must first a take a closer look at what ITACS uses for its current setup. As described by Mike Williams of the ITACS network operations center (NOC), the ITACS unclassified network infrastructure has 139 edge switch stacks on their network and a total of 339 switches configured in these stacks. In each stack, there are between one and eleven switches. Each switch stack acts as one switch. The edge switches being use by ITACS are Ruckus ICX7450 with 48 copper ports (see Figure A.1). The ITACS unclassified network has ten distribution switch stacks that are made of two Ruckus ICX7750 switches with 48 fiber SFP+ ports (see Figure A.2). There are two Extreme MXLe-8 routers (see Figure A.3) in a cluster to form one inside router and two

Extreme MXLe-4 routers (see Figure A.4) in a cluster to form one outside router. There is one Palo Alto PA7050 firewall between the inside and outside routers (see Figure A.5). There are three Dell servers running Windows Server 2012 R2 Datacenter as AD servers, and there are two Dell servers running Window Server 2008 R2 Datacenter as RADIUS servers. ITACS uses Window NPS for RADIUS services [27].

Figure 3.1 shows the ITACS network building diagram of Glasgow Hall. To save time and resources, we chose to only emulate this (Glasgow Hall) network segment, and the emulation includes one Cisco router as the inside router and one Brocade switch (as illustrated in Figure 3.2). The Dell server runs an AD server, a DHCP server, and a RADIUS server. This design only emulates one building of the ITACS network topology. The data that we collected did not require Internet access; therefore, we did not need to establish an Internet connection. ITACS currently runs Windows Server 2012. Because there is a newer version available, Windows Server 2016, we decided to use that version to see if there were any differences/challenges on our smaller test bed. This simplifies by only evaluating the authentication methods and VLAN implementation, which eliminates the need to provide other network security practices implemented by ITACS in their actual network. This streamlines the emulation and speeds up its development and evaluation. Conversely, not implementing the full network configuration may positively or negatively affect the network performance.

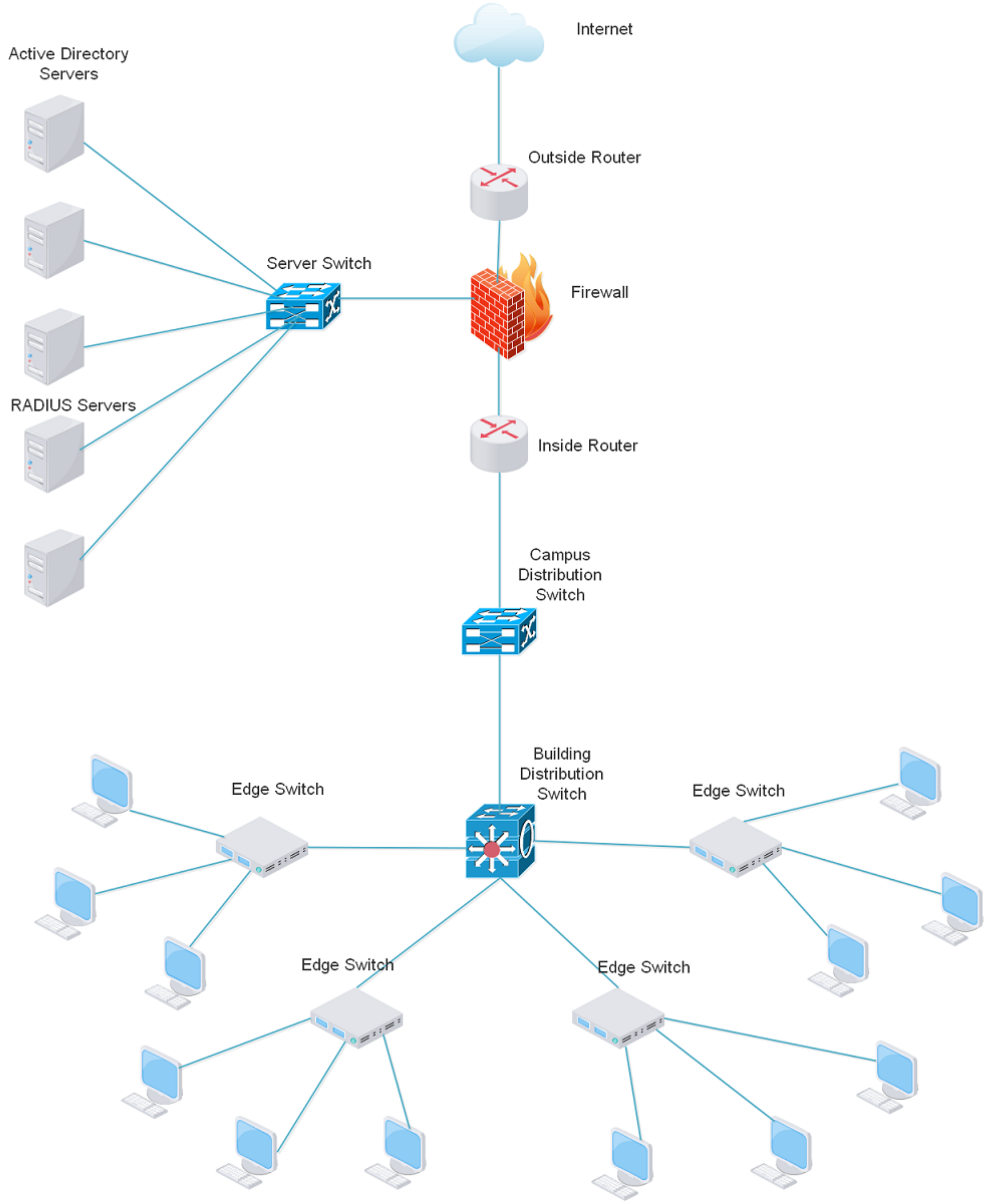


Figure 3.1. Current ITACS Network Topology Showing Details of One Building. Adapted from [27].

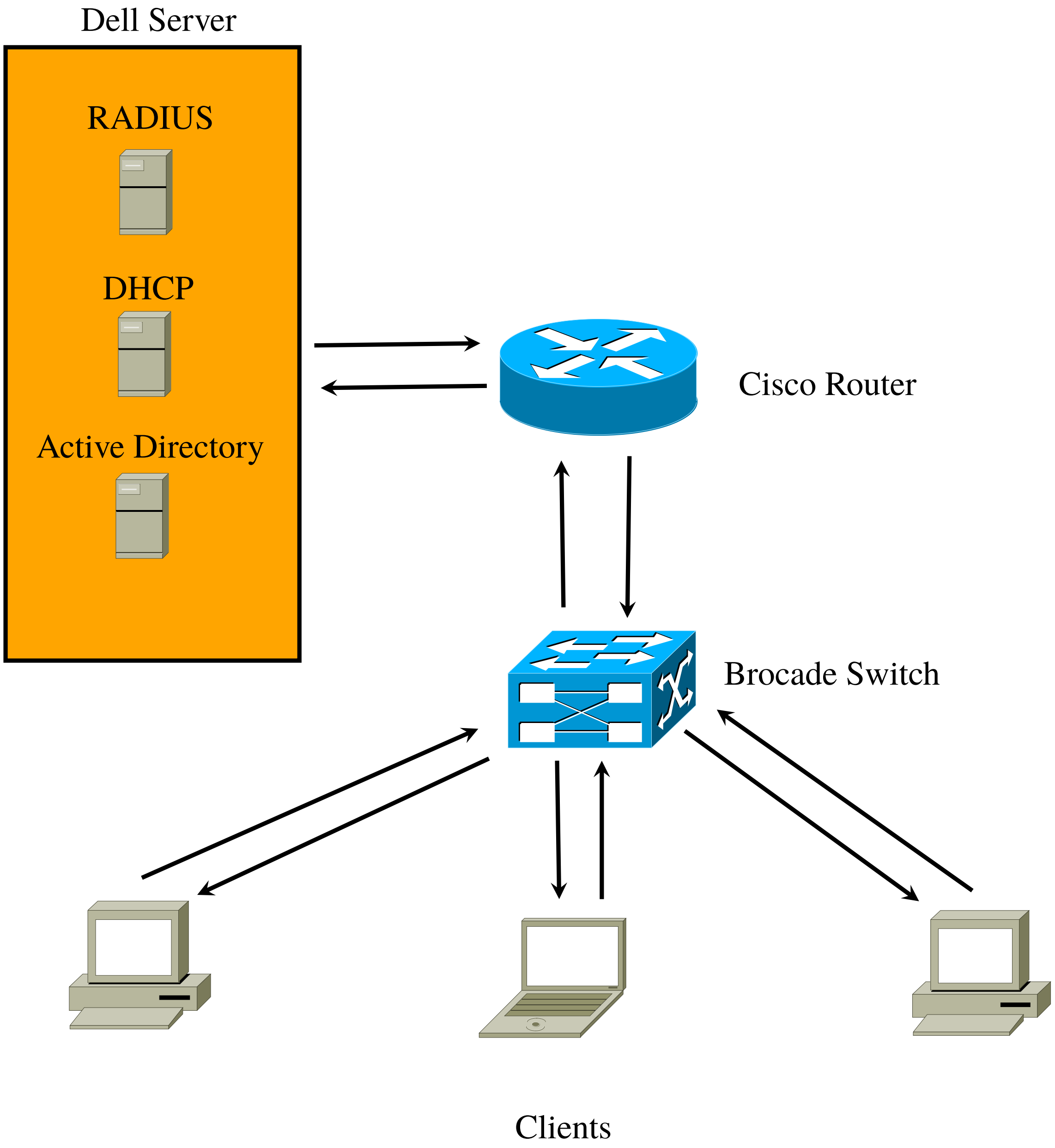


Figure 3.2. ITACS Network Emulation

3.3 Design of SDN Solution

When designing a SDN solution for ITACS, we must consider certain factors that are key in building a network. This thesis focused on robustness, scalability, and cost.

We measure network robustness based on how well a network can maintain functionality when it loses links and nodes or is attacked [28]. We must take into account robustness for ITACS's network because of the number of users, the different types of research, and the critical files and database servers that ITACS hosts. Their network hosts distance learners, on-campus students who are using wireless connections, and the staff connectivity. Robustness is critical.

Scalability is the ability of the network to grow and manage its increased demand [29]. ITACS;s network is based on education and research, which means it must be able to adjust to constant change in the number of users as well as the amount of bandwidth used by each user. Although the number of students and staff is fairly consistent, the number of services and applications needed by students and staff are increasing rapidly. A push for web-based services and applications via the cloud to keep up with demand and scalability is inevitable. SDN architecture provides the resource and storage allocation necessary for cloud-based services to handle the scalability of high demand networks.

Cost is always critical. However, when it comes to Department of Defense (DoD) networks, it is even more important because of budget constraints. Making changes to the network have to be planned carefully within the budget. With networking and cloud technologies evolving at a rapid pace, one needs a network technology that is built with future expansion in mind. Finding ways to build the network with the right equipment and software while not making too many performance sacrifices is the key. SDN excels at providing the capability to grow at a relatively low cost.

For this thesis, the recommendation for ITACS's SDN-based access control solution is to keep the top level of the campus network the same all the way to the campus distribution switch. At the building distribution switch, we place the SDN controller software on a computer/server. The distribution switch connects to the OpenFlow edge switches for hosts/devices to connect to the network.

We chose to treat each building as its own LAN that has its own primary SDN controller that manages the flows and devices on its network, as illustrated in Figure 3.3. This SDN network design allows for management of scalability at the building level. For backup purposes and robustness, each building could use another buildings SDN controller as its secondary controller for failover and load-balancing. This design would also use most of the current network implementation, which lessens the cost of the network upgrade. This solution provides distributed access control for each LAN that is connected to the campus backbone network.

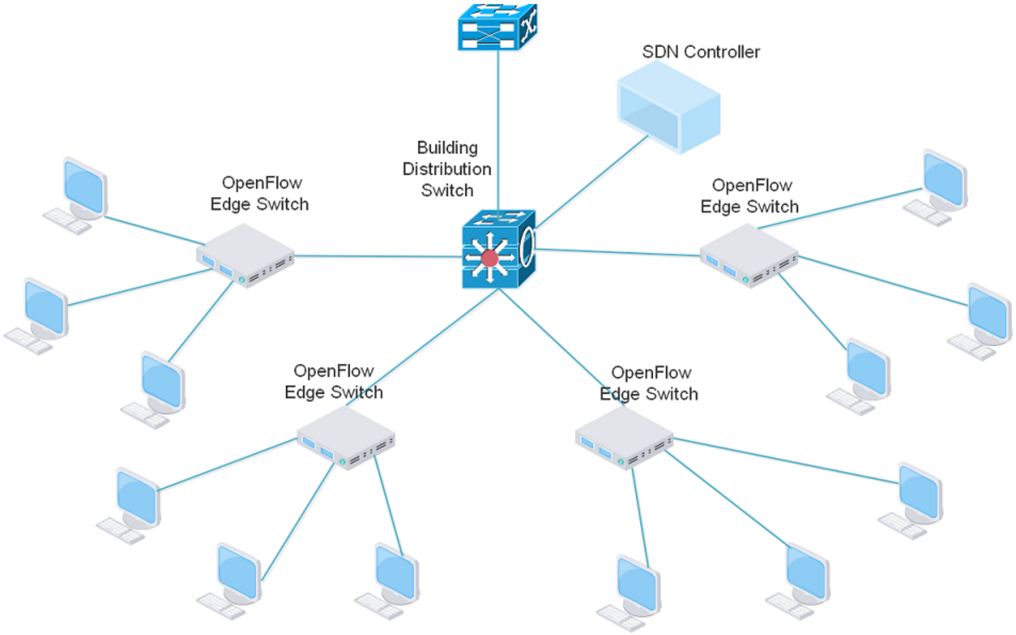


Figure 3.3. Proposed SDN Design: Incremental Deployment at Building Level

To prove that this solution benefits ITACS, without us having to build out an entire building-sized LAN, we decided to concentrate on only the core components of a building LAN. We considered two possible implementations for this experiment.

The first SDN implementation consideration builds on the ITACS implementation that we created. The key element of this implementation plan is to reuse as much as possible from the network created as shown in Figure 3.2. It would require adding on a OpenFlow compatible switch connected to the authentication server and a SDN controller connected to the OpenFlow switch as shown in Figure 3.4. The problems that we ran into were getting compatible switches and SDN controllers within the time frame available. The brocade ICX 7750 switch is OpenFlow enabled, but not easily made compatible with the ONOS SDN controller that we have chosen.

ONOS is one of the leading open source SDN controllers available [30]. ONOS provides the control plane for a SDN to manage network devices and provides software applications and modules that provide communication to hosts and across multiple networks [30]. ONOS has been heavily developed and provides applications for many use cases. One of those use cases is 802.1x authentication with its authorization, authentication and accounting (AAA) application. The AAA application was developed in ONOS version 1.10 for use by Central Offices Re-architected as Datacenters (CORD). CORD is a community-based open source project that is built over ONOS to create telecommunication Central Office datacenter solutions [31].

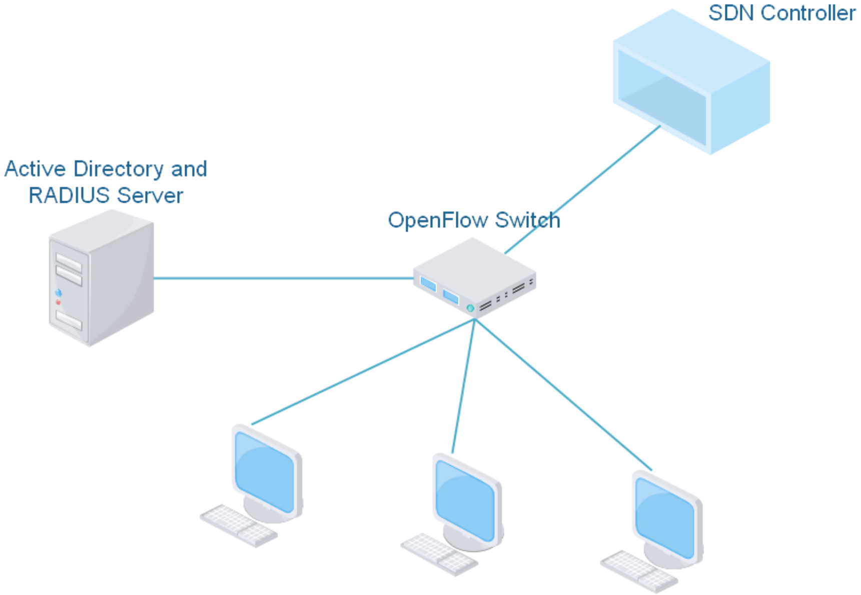


Figure 3.4. First ITACS SDN Implementation Consideration

The second possible design implementation to consider is a partial or simulated implementation of the SDN solution. The pros of this type of implementation are going to be time and cost. With a partial/simulated implementation, we are able to cut time depending on the parts of the network configuration that are simulated. Also, those same considerations in the implementation positively affect the cost. The cons are the exact opposite of the first design consideration: not a full-scale model of the network and the data collected may not be as reliable as it would be in a full-scale SDN operational model. The data analysis has to take into account what has been simulated when presenting results, which in turn affects the decision of moving forward with a full-scale operational implementation.

The partial/simulated implementation was the best direction to go for this thesis due to time constraints and costs being the driving factors [32], [33]. The SDN controller we decided to use was the ONOS controller. We used FreeRadius as the authentication server and MININET to simulate the hosts and OpenFlow switches as shown in Figure 3.5.

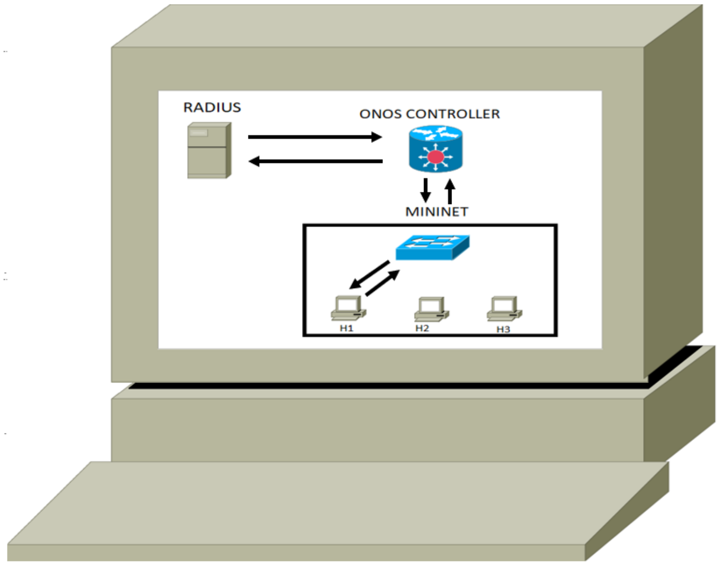


Figure 3.5. Experimental Implementation

3.4 Evaluation Criteria

We evaluate network access control for the ITACS emulation as follows.

- The switch requests the authentication credentials from the host.
- The switch sends the authentication credentials to RADIUS.
- RADIUS verifies hosts credentials for authentication and reply with accept or deny.
- If the authentication is accepted, the host is placed on the authenticated users VLAN

and given an IP address from the DHCP scope of the DHCP server.
- If the authentication fails, the host is placed on the restricted remediation VLAN.

We evaluate network access control for the SDN test environment as follows.

- The switch sends the request to ONOS.
- ONOS sends the authentication request to FreeRADIUS using its AAA application.
- FreeRADIUS verifies hosts credentials for authentication and reply with accept or deny.
- If the authentication is accepted, ONOS adds a flow to the flow table for the host.
- If the authentication fails, ONOS does not add a flow for the host.

We also evaluate the SDN solution based on how other key benefits perform compared to the current ITACS solution. Those features are discussed in the following sections.

3.4.1 Expected Benefits of SDN

Some of the expected benefits of SDN functionality are service speed and agility, flexibility and holistic management, granular security, efficiency(configuration complexity) and lower operating expenses, and lower hardware expenditures [34], [35] .

Service Speed and Agility

Building networks in a SDN architecture using virtual machine (VM)s can be a lot faster that most physical networks. Creating VM instances takes less time than building and adding physical hosts to the network [34].

Holistic Management

On SDN architectures, devices and can be added and removed without the limits that simple network management protocol (SNMP) imposes on networks. This allows for experimentation to be done on your live network without causing degradation of the entire network [34]. Enterprise networks need to add applications and devices on demand for their customers and processing requests such as big data. SDN manages physical and virtual devices from the central controller and also provides a central management console for managing physical and virtual devices via a set of built-in APIs [35].

Granular Security

With networks constantly adding and removing devices, it has become increasingly more difficult to apply firewall and content filtering policies throughout the network. Then, taking into account that most networks allow users to bring their own devices to connect to the network, it creates more security policy distribution issues. SDN, on the other hand, provides a centralized controller that can distribute all network and security policies from a central location. Such centralized control would thus help this increasingly difficult security management problem [34], [35].

Configuration Complexity

Configuration complexity can be evaluated by comparing how SDN is built to have centralized configuration and control from the SDN controller for the entire network. Most network configurations may be performed and even automated on a small number of controllers. The devices on the forwarding plane receive all their instructions from the controller. You can add or remove devices without any more configurations being needed. This is one of the primary features of SDN architecture that make it so appealing to replace current network solutions. Right now, ITACS has to configure each device individually when they add a new device (e.g., switch, router, or firewall) to the network. Adding new devices may also cause ITACS to have to change some configurations on their current devices as well because of vendor specific compatibility issues.

Hardware and Operational Costs

The current ITACS solution uses one vendor, Brocade, for their network devices. Being locked in to a vendor because of vendor-specific features and the vendor not being easily compatible with other vendors, makes it hard to make choices based on cost instead of convenience. With SDN architecture, you can go with the cheapest available devices because compatibility is not an issue. You can also use open source software to create virtual switches. Many SDN controllers are open source as well, which cuts hardware costs dramatically during implementation.

CHAPTER 4:
Experimentation

In this chapter, we explain what hardware and software configuration choices were made and why. Then we present the main experimental results of the thesis. The configuration details of the experimentation can be viewed in Appendix A.3.

4.1 ITACS Implementation

The ITACS experimental setup includes one Dell server, one Cisco router, three Brocade switches and three Dell workstations. For this setup to work, we made sure that the server, router, and switches were all on the same VLAN.

ITACS Experimental Solution		
Hardware	Software	Role
Dell GCNVX12 Server	Windows Server 2016	DC, DNS, CA, DHCP, NPS
Dell Optiplex 9020	Windows 10	clients
Cisco Router	Cisco 2811	Distribution Switch
Brocade Switches	Brocade ICX 6450	Edge Switches

4.1.1 Windows Server

On our Dell GCNVX12 Server, we have installed Windows Server 2016. As part of the Windows Server suite, we have installed AD as our user database provider and DC, NPS to use as RADIUS, DHCP for IP address assignment and CA to provide certificates for authentication.

Active Directory

In AD, we need to create our security groups for authenticated users and computers. These have to be defined for us to add user accounts to these security groups so we can create network policies in NPS for the authentication and network resource access purposes [12].

Certificate Authority

We needed AD CS certification authority (CA) installed to automatically enroll a server certificate to all of our NPS servers and to send client computers CA certificates. These CA certificates are stored on each client computer and in the Root Trusted Certificate Authorities store. The certificate allows the client computer to perform 802.1x authentication by sending its certificate to the edge switch, then sends the certificate to NPS server for verification to see if the computer is trusted [21], [12].

Dynamic Host Configuration Protocol

We decided to use DHCP on the server instead of using the cisco router because of the centralized configuration and maintenance capabilities. We also chose Windows Server for this role because it simplifies the configuration process and ITACS uses the same configuration setup.

Network Policy Server

We are deploying NPS as RADIUS server. The brocade switches are configured as RADIUS clients. NPS is configured to talk to the switches to authenticate client computers. A connection request policy is created to allow the RADIUS clients to talk to the RADIUS server [21]. Then, we configure Network Policies that include more details for how to authenticate users and computers. We configure our Network Policies to use certificate based PEAP authentication for 802.1x and PAP for MAC authentication [12]. We followed the following steps from Implementing IEEE 802.1x for Wired Networks paper by Johan Loos [21]:

4.1.2 VLANs

For our ITACS solution, we have two VLANs, a restricted VLAN for users who fail to authenticate and a VLAN for authenticated users and devices.

VLAN ID	Description	location
2	Native VLAN	Distribution switch
308	Authenticated user VLAN	Edge switches
222	Restricted VLAN	Edge switches

Native VLAN

On the Cisco router we configured VLAN 2 as the native VLAN with trunk mode encapsulation Enabled and to allow all VLANs for the four FastEthernet interfaces (fa 0/0/0-fa 0/0/3). This configuration allows un-tagged passing through the router interfaces to tagged as VLAN 2 traffic since that is the native VLAN, but it allows all other traffic tagged for any other VLAN to pass through as well. The following configuration was used to create the native VLAN and to configure trunk mode on the four interfaces [36].

```
router>en
router#vlan database
router(vlan)#vlan 2
VLAN 2 added:
Name:VLAN0002
router(vlan)#exit
APPLY completed
Exiting...
router#config t
router(config)#int fa 0/0/0
router(config-if)#switchport mode trunk
router(config-if)#switchport trunk encapsulation dot1q
router(config-if)# switchport trunk native vlan 5
router(config-if)# switchport trunk allow vlan all
router(config-if)#exit
router(config)#
...
```

Authenticated and Restricted VLAN

First, let us discuss how to configure VLANs on brocade switches. Similar to Cisco switches, we have to declare/initialize VLANs individually. Where things start to differ is when we get to how we assign interfaces to the VLANs. Instead of going to each interface and tagging the VLAN, we go to the VLAN and tag or un-tag the interfaces. When we un-tag an interface to a VLAN, that means all un-tagg edged traffic that comes through that interface gets tagged to that VLAN by default. If we tag an interface to a specific VLAN, that means that VLAN is allowed on that interface. An interface can be tagged by multiple VLANs, but can only be un-tagged by one VLAN. To allow both tagged and un-tagged traffic over an interface we have to configure the interface for dual-mode, which is the equivalent of putting an interface in trunk mode with a native VLAN on cisco devices. We configured our interface 1/1/48 for dual-mode on VLAN 308 and tagged on VLAN 222. The following code was used to configure the VLANs on the brocade switches:

```
switch>en
switch#config t
switch(config)#vlan 308
switch(config-vlan-308)#tag int eth 1/1/48
switch(config-vlan-308)#exit
switch(config)# int eth 1/1/48
switch(config-if-e1000-1/1/48)#dual-mode 308
switch(config-if-e1000-1/1/48)#exit
switch(config)#vlan 222
switch(config-vlan-222)#tag eth 1/1/48
switch(config-vlan-222)#exit
switch(config)#exit
switch#
...
```

We configure flex authentication on the Brocade switches. This means that MAC and 802.1x authentication are configured on the switches. MAC authentication takes precedence over 802.1x authentication. If a host's MAC is accepted, then 802.1x does not get used. If MAC authentication fails, the host gets prompted to enter credentials for 802.1x authentication. If host/user is able to authenticate, they are placed on VLAN 308. If a user fails to authenticate with 802.1x authentication, the host is put into the restricted VLAN. The switch configurations for this authentication method are shown below. All other switch configurations are located in the appendix section.

```
switch>en
switch#config t
switch(config)#authentication
switch(config-authen)#auth-order mac-auth dot1x
switch(config-authen)#auth-default-vlan 308
switch(config-authen)#restricted-vlan 222
switch(config-authen)#auth-fail-action restricted-vlan
switch(config-authen)#re-authentication
switch(config-authen)#auth-vlan-mode multiple-untagged
switch(config-authen)#dot1x enable
switch(config-authen)#dot1x enable eth 1/1/1 to 1/1/47
switch(config-authen)#mac-authentication enable
switch(config-authen)#mac-authentication enable eth 1/1/1 to 1/1/47
switch(config-authen)#mac-authentication password-override 1 *********
switch(config-authen)#mac-authentication dot1x-override
switch(config-authen)#exit
switch(config)#aaa authentication dot1x default radius
switch(config)#aaa authentication login default radius local
switch(config)#aaa authentication login privilege-mode
switch(config)#radius-server host 172.20.42.110 auth-port 1812 acct-port 1813 authentication-only key
2 JVFHbysxVTg6OEdWZzhA dot1x
switch(config)#exit
switch#
...
```

4.1.3 Experiment Scenarios

Now that we have implemented our ITACS experimental solution, we have to test the network to make sure it is fully functional. The following scenarios are to validate that the ITACS experimental solution is performing the network access control configurations as intended.

Scenario One

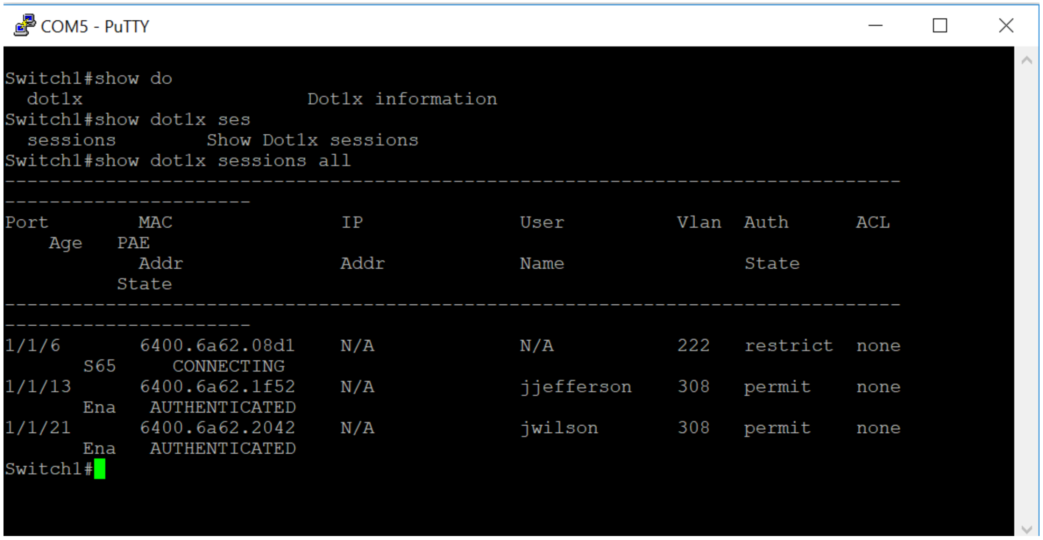
A client computer fails MAC authentication and is trying to connect using 802.1x authentication.

Scenario Two

A client computer trying to authenticate and fails MAC authentication and 802.1x authentication. Consequently, the client computer is placed on the restricted VLAN.

4.1.4 Results

After running through both scenarios, we see that both were successful from the screenshot of the Switch1 (shown in Figure 4.1).

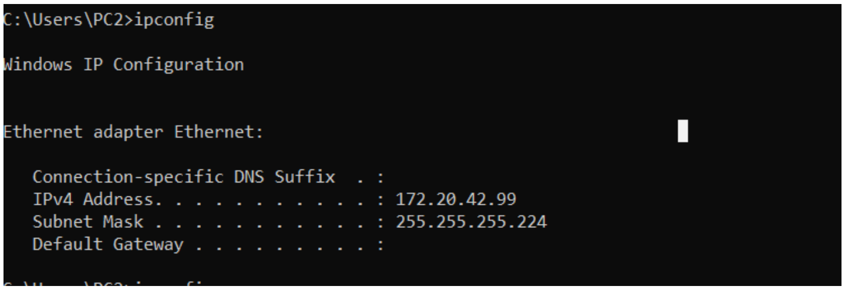


```
Switch1#show do
  dot1x                    Dot1x information
Switch1#show dot1x ses
  sessions         Show Dot1x sessions
Switch1#show dot1x sessions all
--------------------------------------------------------------------------------
----------------------
Port        MAC                 IP              User          Vlan  Auth      ACL
    Age    PAE
            Addr                Addr            Name                State
          State
--------------------------------------------------------------------------------
----------------------
1/1/6       6400.6a62.08d1      N/A             N/A           222   restrict  none
       S65     CONNECTING
1/1/13      6400.6a62.1f52      N/A             jjefferson    308   permit    none
       Ena    AUTHENTICATED
1/1/21      6400.6a62.2042      N/A             jwilson       308   permit    none
       Ena    AUTHENTICATED
Switch1#
```

Figure 4.1. Screenshot of Edge Switch Showing VLAN Placement after Authentication Attempts

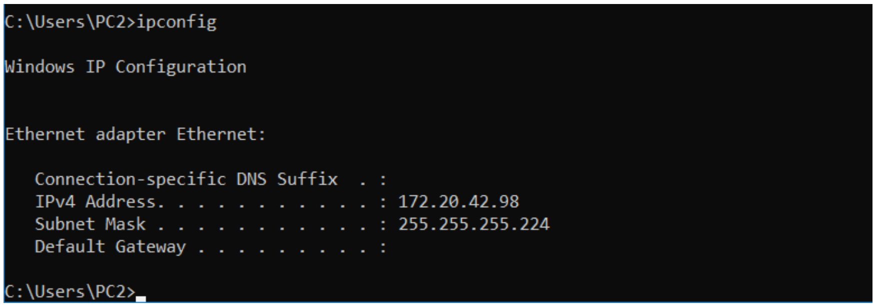
Scenario One Results

Two client computers, PC1 and PC2, successfully authenticated via 802.1x are showing that they are on the appropriate VLAN for authenticated users, 308 (as shown in Figure 4.1). PC1 and PC2 have received their IP addresses from the DHCP scope provided by our DHCP server (as shown in Figures 4.2 and 4.3).



```
C:\Users\PC2>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 172.20.42.99
   Subnet Mask . . . . . . . . . . . : 255.255.255.224
   Default Gateway . . . . . . . . . :
```

Figure 4.2. IP Address Assigned by DHCP to PC1

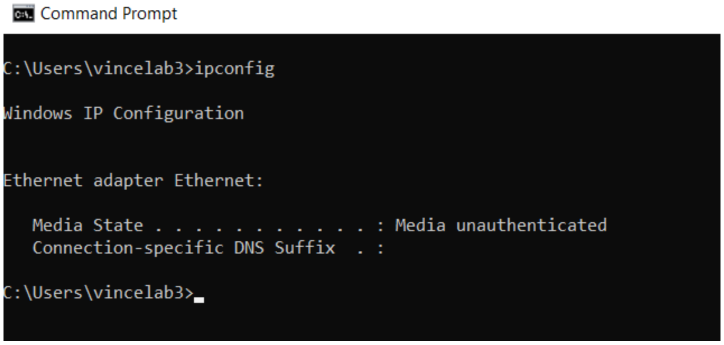


```
C:\Users\PC2>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   IPv4 Address. . . . . . . . . . . : 172.20.42.98
   Subnet Mask . . . . . . . . . . . : 255.255.255.224
   Default Gateway . . . . . . . . . :

C:\Users\PC2>
```

Figure 4.3. IP Address Assigned by Dynamic Host Configuration Protocol (DHCP) to PC2

Scenario Two Results

We can see that when the client computer failed authentication, it was placed on the restricted VLAN 222 as expected (as shown in Figure 4.1). Because the computer was placed in the restricted VLAN, it was assigned an IP address (as shown in Figure 4.4).



```
C:\Users\vincelab3>ipconfig

Windows IP Configuration


Ethernet adapter Ethernet:

   Media State . . . . . . . . . . . : Media unauthenticated
   Connection-specific DNS Suffix  . :

C:\Users\vincelab3>
```

Figure 4.4. No IP Address Assigned to PC3

Configuration Complexity

The ITACS test implementation required over two hundred lines of configuration code for each of the three switches and about a hundred lines of configuration code for the router. With Brocade switches, they can be configured in stacks. This means that we only have to manually configure the master switch in the stack and the other switches automatically copy those configurations. There are 139 switch stacks on the ITACS unclassified network [27].

$$SwitchStacks * LinesOfCode = TotalLinesOfCode$$

$$139 * 200 = 27,800 \tag{4.1}$$

That means the administrator would have to manually input about 30,000 lines of configurations in across many different locations which takes a lot of time as well.

Operational Costs
As stated in Section 3.2, there are 339 switches on ITACS network. Those switches are either a Brocade ICX 7750 ($2,795.00 used to $13,985.49 brand new) or 7450 ($245.97 used to $7,137.40 brand new) [37], [38], [39], [40]. The cost of the switches for their network could have cost them up to $4,741,243.83.

4.2 SDN Implementation

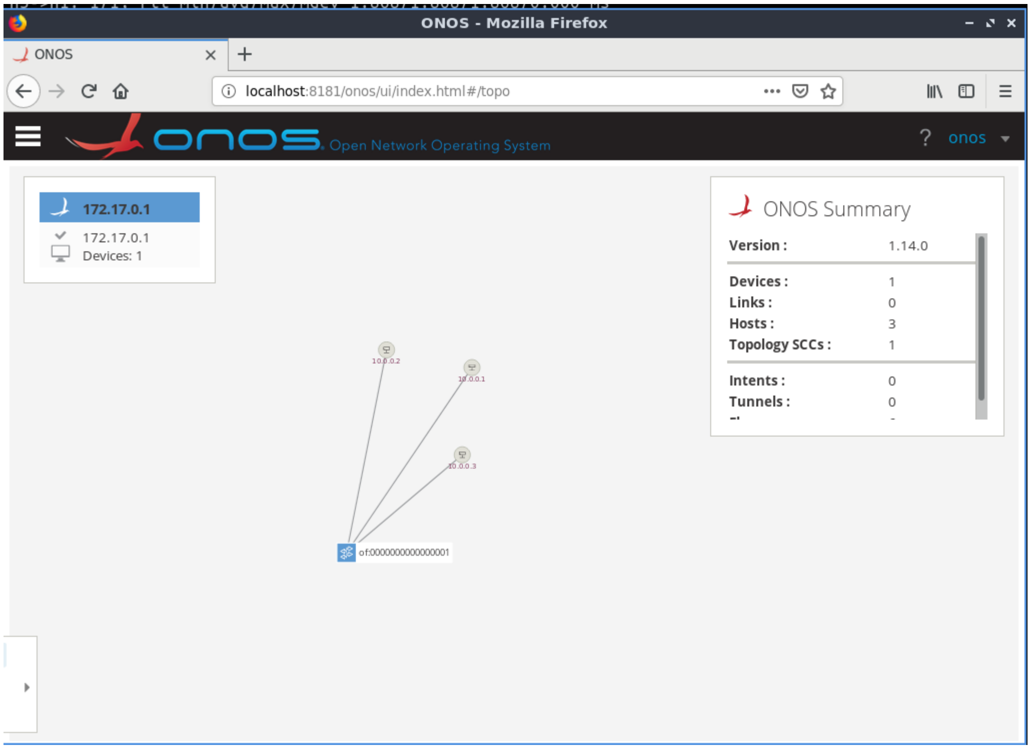
The implementation that we are implementing is from a case study done by ONOS developers using the AAA application from CORD to negotiate authentication requests with FreeRADIUS as the authentication server [31]. Although we did configure the authentication for this implementation, we are only testing ONOS for completeness of functionality, configuration complexity and operational costs to see if the SDN solution is a good fit to replace ITACS's current network solution. All the installation and configuration details can be viewed in Appendix A.4.

4.2.1 Experimental Scenario

We built a SDN with switch and three hosts in MININET connected to a remote ONOS controller. From this build, we are able to view the complete network topology from ONOS's topology management view feature. From there we can view the low-level management information such as MAC addresses, port numbers and VLANs.

4.2.2 Results

The SDN was successfully built with MININET and ONOS. We view the topology in the web browser (as shown in Figure 4.5). This is a demonstration of the holistic management feature of ONOS displaying the network devices that are currently connected.



ONOS Summary

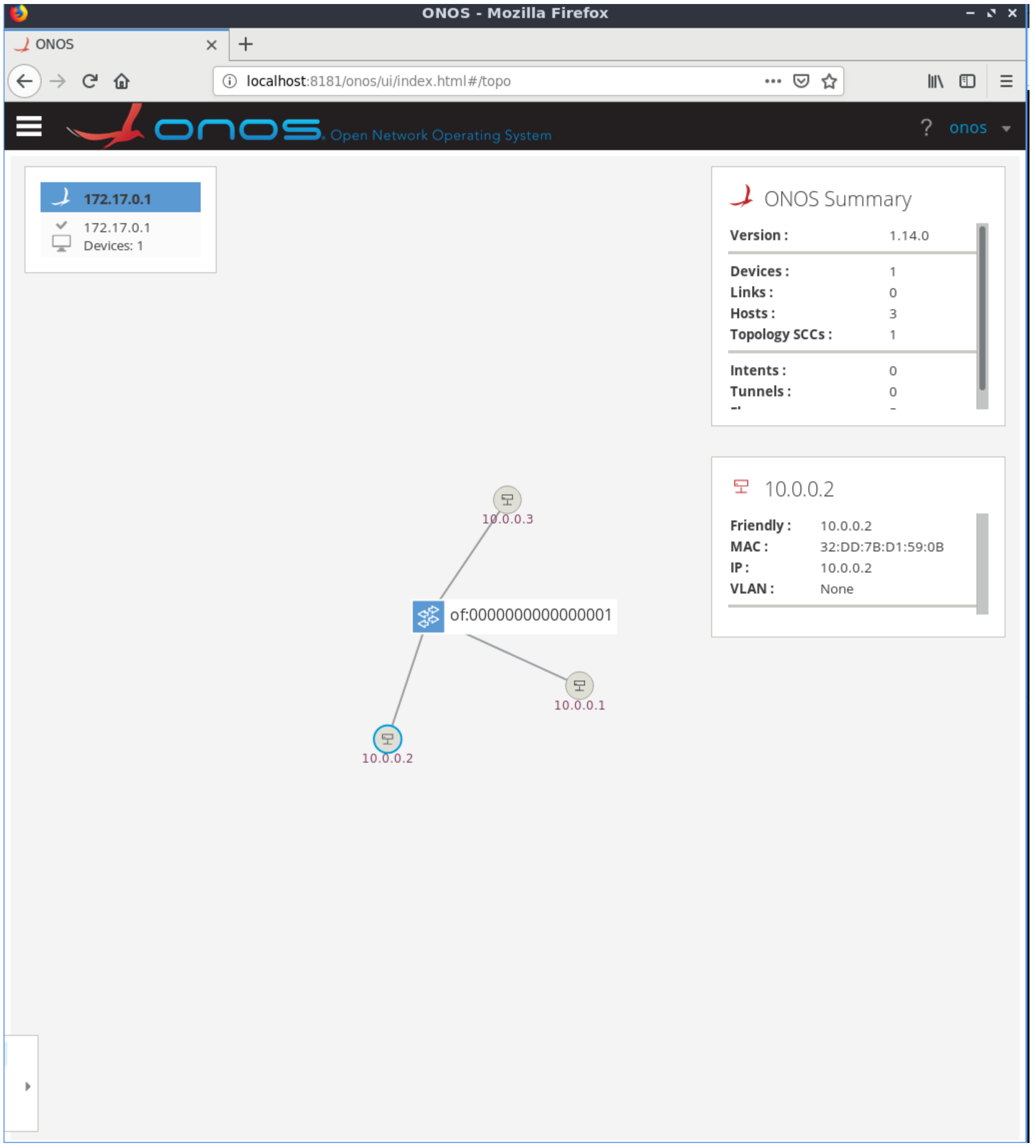
Version : 1.14.0

Devices : 1
Links : 0
Hosts : 3
Topology SCCs : 1

Intents : 0
Tunnels : 0

Figure 4.5. ONOS Topology View

From this same view, we can view the MAC and IP addresses by clicking on the host device as shown in Figure 4.6. We can also see what VLAN the device is on if VLANs are configured. We can also see the port the device is connected on the switch when we click on the link between the two devices as shown in Figure 4.7.



172.17.0.1

172.17.0.1
Devices: 1

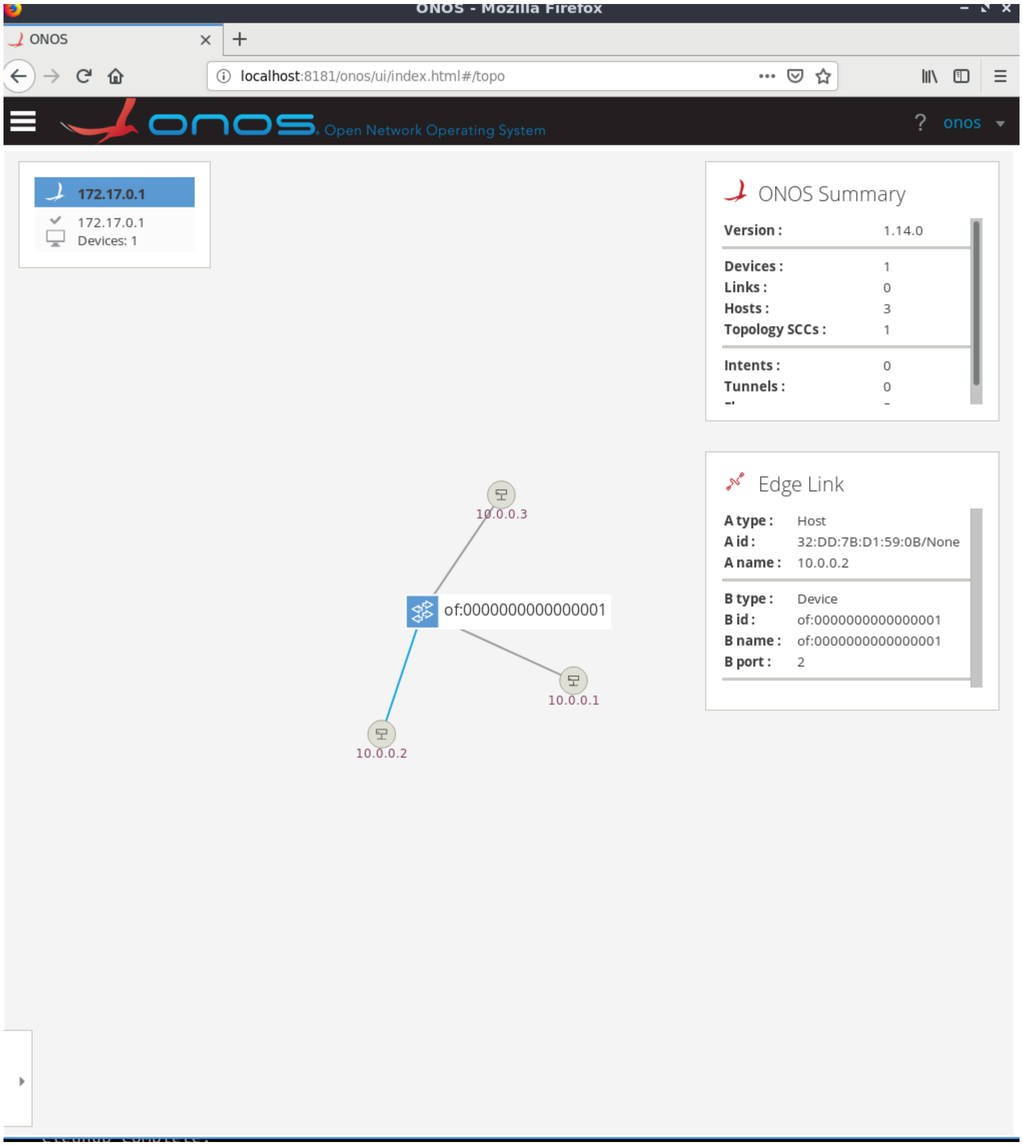
ONOS Summary

Version :	1.14.0
Devices :	1
Links :	0
Hosts :	3
Topology SCCs :	1
Intents :	0
Tunnels :	0

10.0.0.2

Friendly :	10.0.0.2
MAC :	32:DD:7B:D1:59:0B
IP :	10.0.0.2
VLAN :	None

Figure 4.6. Low-Level Management Information from Host



ONOS Summary

Version :	1.14.0
Devices :	1
Links :	0
Hosts :	3
Topology SCCs :	1
Intents :	0
Tunnels :	0

Edge Link

A type :	Host
A id :	32:DD:7B:D1:59:0B/None
A name :	10.0.0.2
B type :	Device
B id :	of:0000000000000001
B name :	of:0000000000000001
B port :	2

Figure 4.7. Low-Level Management Information from Link

Configuration Complexity

Configuring the SDN solution took much less time than the ITACS network solution. Because the virtual OpenFlow switch on the SDN solution receives all forwarding information from SDN controller, ONOS, and its applications, we only had to do configurations once. No matter how many OpenFlow switches we add to the SDN solution, the complexity does not increase.

Hardware Costs

One of the best parts about a SDN solution is that a lot of the software is free and open source. This means we can reuse hardware that we already have available and download the compatible SDN software for the switches and the controllers free of additional costs. There are virtual OpenFlow switch solutions available as well so we don't have to use relatively expensive physical switches in our implementation of a SDN solution.

CHAPTER 5: Conclusions and Future Work

In this chapter, we present our conclusions from this thesis research and outline several areas of future work.

5.1 Conclusions

The current VLAN based network access control used by ITACS cannot keep track of all devices connected to the network, which allows unauthorized devices to possibly be connected and go unnoticed. We experimented with SDN technology to determine if it provided a more efficient solution to this problem. The results show that SDN not only is able to maintain the same level of network access control but also provides network administrators an accurate view of device level topology. For any network device connecting to a SDN, the controller will have knowledge of and therefore be able to produce an accurate automated topology view of the entire LAN. This decreases management overhead significantly.

The SDN solution's configuration complexity compared to the current ITACS solution would save network administrators and operators a lot of working hours when switches and other devices need to be added or removed. This type of configuration is possible because SDN's centralized programming design provides the flexibility to increase or decrease in size as needed without more configuration at the forwarding plane level. The availability of open source SDN software may also help lower operational costs and hardware costs. Network administrators can also lower hardware costs by choosing the lowest cost switches available to implement their SDN architecture without being trapped into vendor proprietary switches. This does depend on the on the SDN controller chosen by the network administrator.

In summary, the findings of this thesis suggest that SDN would clearly be a good choice for ITACS.

5.2 Future Work

SDN controller security should be considered a priority when using open source software in a network operations environment. Network intruders have access to the same software we do and can easily take down or take control of the network if they are able to get to the controller. Building a full SDN solution would provide us with the results to compare both network implementations properly. More time to plan and select the right combination of SDN controller, switches and software would make for a smoother install and implementation. From the research and troubleshooting, we learned that this is still a very new technology and some vendors are not compatible or simple to use with certain SDN controllers. Doing more research to figure out what devices work well together or what requirements for the devices to work well together ahead of time would fix a lot of these problems that we ran into.

APPENDIX: Hardware and Software Configuration Details

A.1 Hardware

Figure A.1. Ruckus ICX7450 48 Port Switch. Source: [1].

Figure A.2. Ruckus ICX7750 48 Fiber Port Switch

Figure A.3. Extreme MXLe-8 Router. Source: [2].

Figure A.4. Extreme MXLe-4 Router. Source: [2].

Figure A.5. Palo Alto PA7050 Firewall. Source: [3].

A.2 Switch Configuration

```
Switch1#show configuration
!
Startup-config data location is flash memory
!
Startup configuration:
!
ver 08.0.30qT311
!
stack unit 1
  module 1 icx6450-48-port-management-module
  module 2 icx6450-sfp-plus-4port-40g-module
stack disable
stack mac 748e.f8b7.4120
!
!
!
!
vlan 222 name Remediation by port
 tagged ethe 1/1/48
!
vlan 254 name DEFAULT-VLAN by port
 no spanning-tree
!
vlan 301 name ADMIN by port
 tagged ethe 1/1/48
!
vlan 308 name MAC-AUTH by port
```

```
 tagged ethe 1/1/48
!
!
!
!
authentication
 auth-order mac-auth dot1x
 auth-default-vlan 308
 restricted-vlan 222
 auth-fail-action restricted-vlan
 re-authentication
 auth-vlan-mode multiple-untagged
 dot1x enable
 dot1x enable ethe 1/1/1 to 1/1/47
 mac-authentication enable
 mac-authentication enable ethe 1/1/1 to 1/1/47
 mac-authentication password-override 1 *********
 mac-authentication dot1x-override
!
aaa authentication web-server default radius local
aaa authentication dot1x default radius
aaa authentication login default radius local
aaa authentication login privilege-mode
jumbo
default-vlan-id 254
enable acl-per-port-per-vlan
hostname Switch1
ip icmp burst-normal 5000 burst-max 10000 lockup 300
ip tcp burst-normal 10 burst-max 100 lockup 300
ip address 172.20.42.107 255.255.255.224
ip dns domain-list VinceLab.vincelab.com
no ip dhcp-client enable
ip multicast passive
ip default-gateway 172.20.40.1
!
password-change console-cli
radius-server host 172.20.42.110 auth-port 1812 acct-port 1813 authentication-only key 2 *************** dot1x
!
!
clock timezone us Alaska
hitless-failover enable
interface ethernet 1/1/1
 dot1x port-control auto
 dhcp snooping trust
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/2
 dot1x port-control auto
 dhcp snooping trust
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/3
```

```
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/4
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/5
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/6
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/7
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/8
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/9
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/10
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/11
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/12
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/13
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/14
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/15
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/16
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/17
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/18
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/19
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/20
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/21
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/22
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/23
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
```

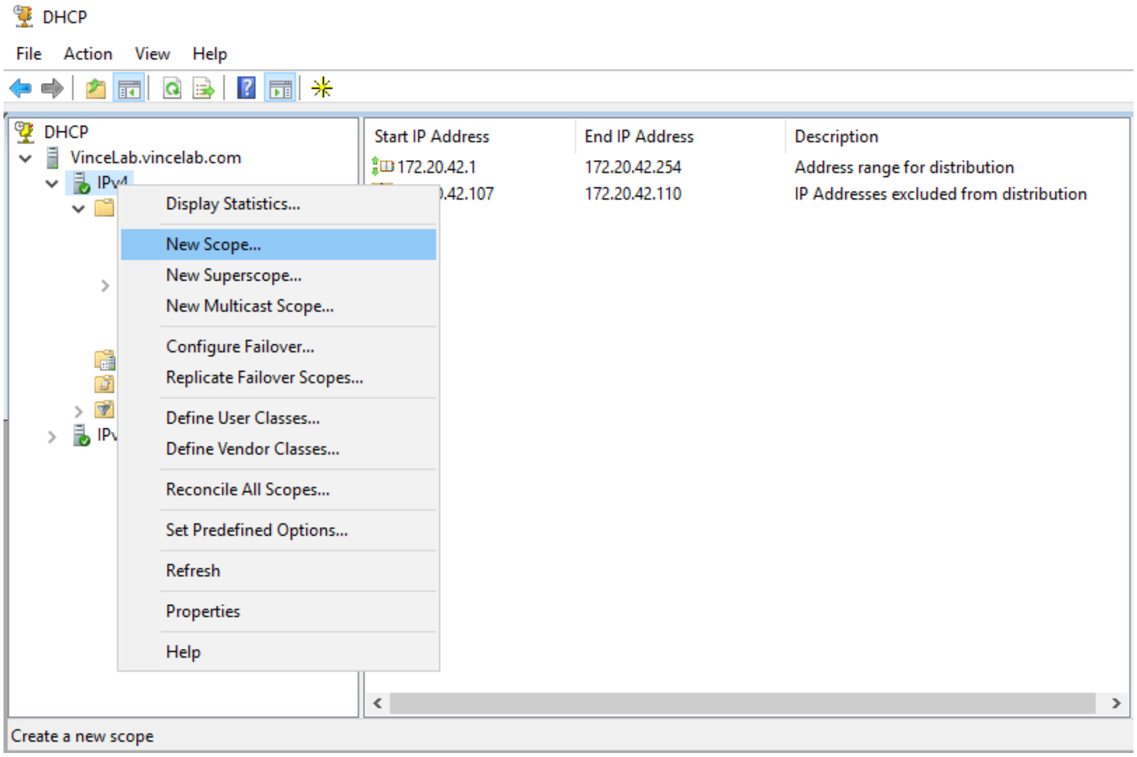
```
interface ethernet 1/1/24
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/25
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/26
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/27
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/28
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/29
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/30
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/31
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/32
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/33
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/34
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/35
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/36
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/37
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/38
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/39
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/40
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/41
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/42
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/43
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/44
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
```

```
!
interface ethernet 1/1/45
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/46
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/47
 authentication max-sessions 7
 dot1x port-control auto
 spanning-tree 802-1w admin-edge-port
!
interface ethernet 1/1/48
 dual-mode  308
 no spanning-tree
!
interface ethernet 1/2/2
 speed-duplex 1000-full-master
!
interface ethernet 1/2/4
 speed-duplex 1000-full-master
!
!
!
!
!
!
!
!
end


...
```

A.3 ITACS Implementation Setup

This is how we setup DHCP:

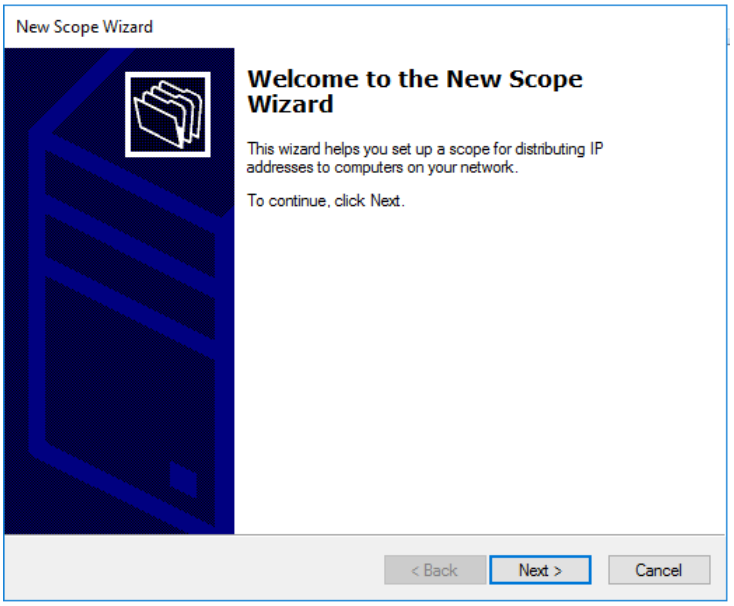
- Open DHCP Console from Administrative Tools, right click on IPv4 and select New Scope



Start IP Address	End IP Address	Description
172.20.42.1	172.20.42.254	Address range for distribution
.42.107	172.20.42.110	IP Addresses excluded from distribution

Create a new scope

- On the Welcome to the New Scope Wizard page, click Next

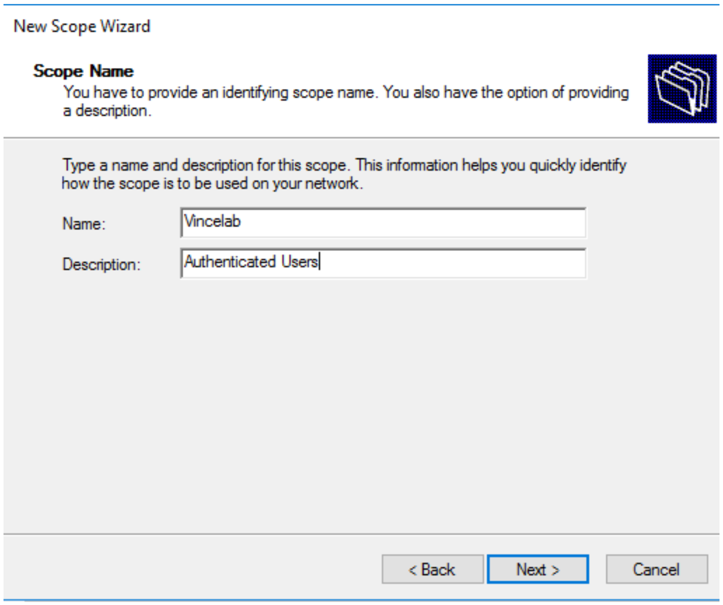


Welcome to the New Scope Wizard

This wizard helps you set up a scope for distributing IP addresses to computers on your network.

To continue, click Next.

- On the Scope Name page, type a name for the scope and click Next


New Scope Wizard

Scope Name

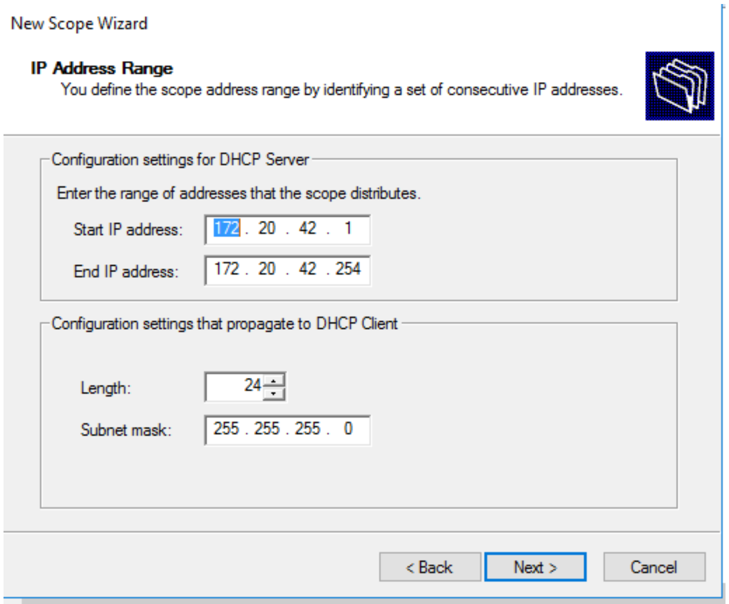
You have to provide an identifying scope name. You also have the option of providing a description.

Type a name and description for this scope. This information helps you quickly identify how the scope is to be used on your network.

Name: Vincelab

Description: Authenticated Users

- On the IP Address Range page, specify Start and End IP address. Also specify the correct subnet mask and click Next


New Scope Wizard

IP Address Range

You define the scope address range by identifying a set of consecutive IP addresses.

Configuration settings for DHCP Server

Enter the range of addresses that the scope distributes.

Start IP address: 172 . 20 . 42 . 1

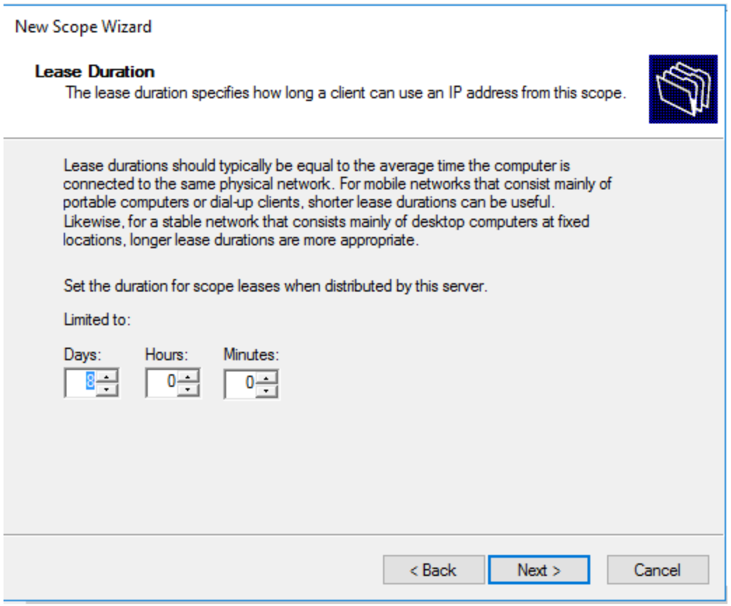
End IP address: 172 . 20 . 42 . 254

Configuration settings that propagate to DHCP Client

Length: 24

Subnet mask: 255 . 255 . 255 . 0

- On the Add Exclusions page, click Next
- On the Lease Duration page, specify a lease duration and click Next



New Scope Wizard

Lease Duration

The lease duration specifies how long a client can use an IP address from this scope.

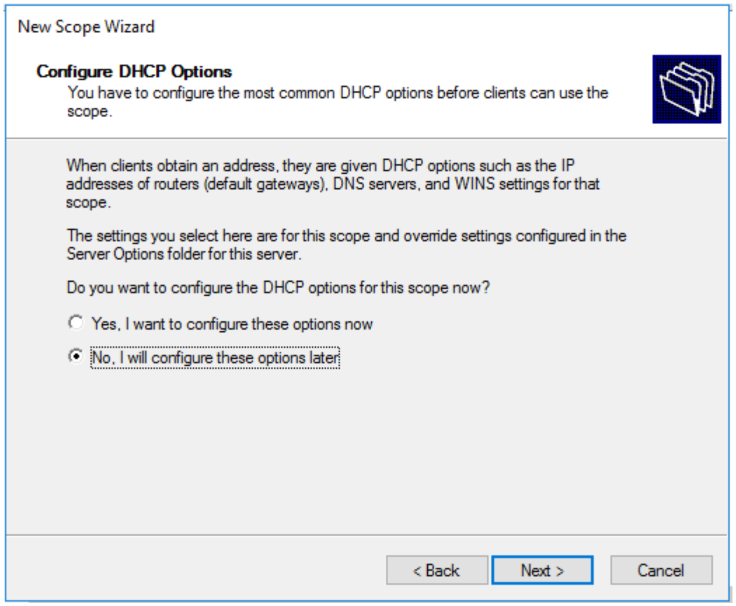
Lease durations should typically be equal to the average time the computer is connected to the same physical network. For mobile networks that consist mainly of portable computers or dial-up clients, shorter lease durations can be useful. Likewise, for a stable network that consists mainly of desktop computers at fixed locations, longer lease durations are more appropriate.

Set the duration for scope leases when distributed by this server.

Limited to:

Days: 8 Hours: 0 Minutes: 0

- On the Configure DHCP Option page, select No, I will configure these options later and click Next



New Scope Wizard

Configure DHCP Options

You have to configure the most common DHCP options before clients can use the scope.

When clients obtain an address, they are given DHCP options such as the IP addresses of routers (default gateways), DNS servers, and WINS settings for that scope.

The settings you select here are for this scope and override settings configured in the Server Options folder for this server.

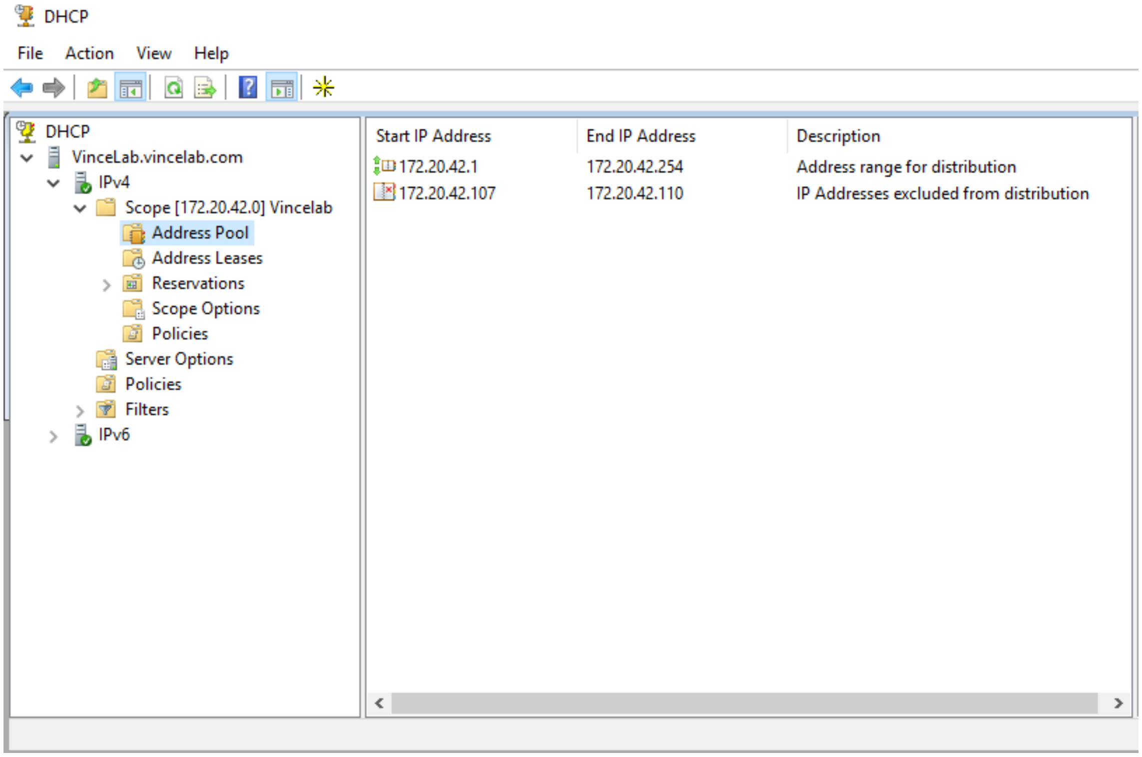
Do you want to configure the DHCP options for this scope now?

○ Yes, I want to configure these options now

● No, I will configure these options later

- On the Completing the New Scope Wizard page, click Finish

We created one scope (172.02.42.0/24) that excluded the IP addresses of the servers, routers and switches.



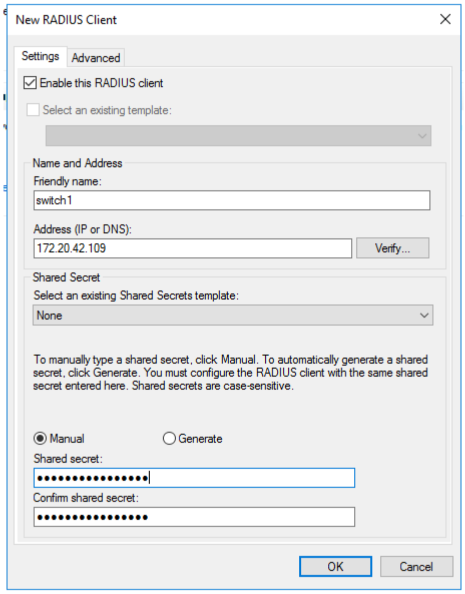
Start IP Address	End IP Address	Description
172.20.42.1	172.20.42.254	Address range for distribution
172.20.42.107	172.20.42.110	IP Addresses excluded from distribution

Network Policy Server

We are deploying NPS as RADIUS server. The brocade switches are configured as RADIUS clients. NPS will be configured to talk to the switches to authenticate client computers. A connection request policy is created to allow the RADIUS clients to talk to the RADIUS server [21]. Then, we configure Network Policies that include more details for how to authenticate users and computers. We configure our Network Policies to use certificate-based PEAP authentication for 802.1x and PAP for MAC authentication [12]. We followed the following steps from Implementing IEEE 802.1x for Wired Networks paper by Johan Loos [21]:

Configure RADIUS Client on NPS Server

- Open Network Policy Server from Administrative Tools, expand RADIUS Clients and Servers, right click on RADIUS Clients
- On the New RADIUS Client dialog box, specify a friendly name and IP address
- Specify a Shared Secret



New RADIUS Client

Settings | Advanced

☑ Enable this RADIUS client

□ Select an existing template:

Name and Address

Friendly name: switch1

Address (IP or DNS): 172.20.42.109

Shared Secret

Select an existing Shared Secrets template: None

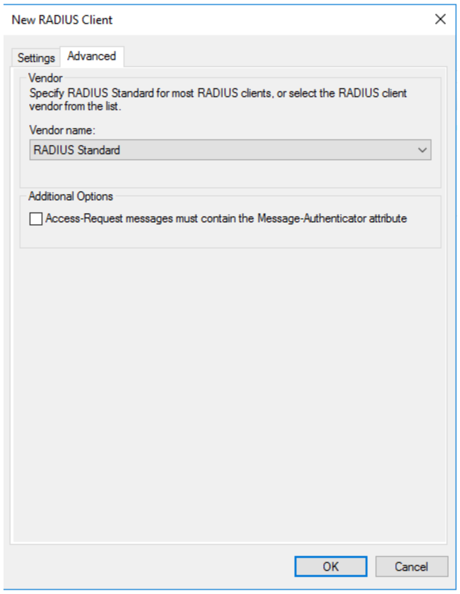
To manually type a shared secret, click Manual. To automatically generate a shared secret, click Generate. You must configure the RADIUS client with the same shared secret entered here. Shared secrets are case-sensitive.

● Manual ○ Generate

Shared secret: ••••••••••••••••

Confirm shared secret: ••••••••••••••••

- Click on Advanced, uncheck or check the required options



New RADIUS Client

Settings | Advanced

Vendor

Specify RADIUS Standard for most RADIUS clients, or select the RADIUS client vendor from the list.

Vendor name: RADIUS Standard

Additional Options

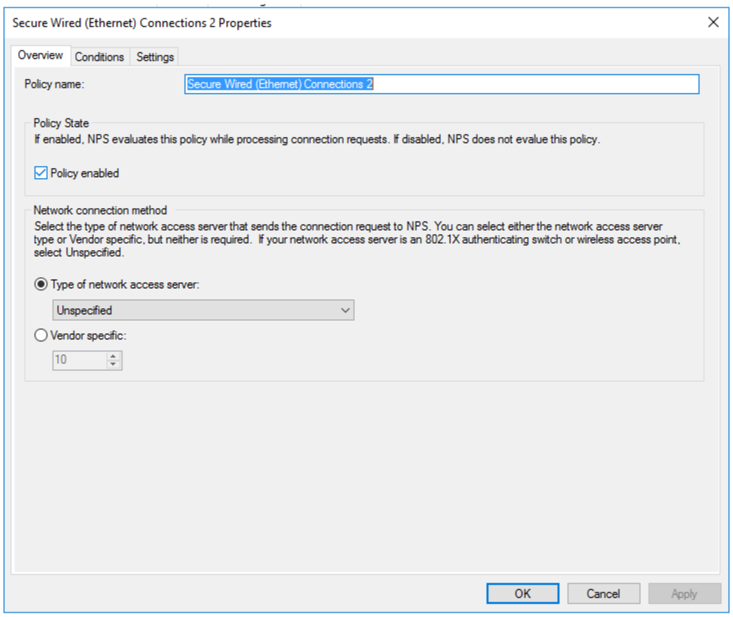
□ Access-Request messages must contain the Message-Authenticator attribute

- Click OK

Configure Connection Request Policy

- From the Network Policy Server Console, right click on Connection Request Policies and select New

- On the Specify Connection Request Policy Name and Connection Type page, type a name for the policy and click Next



Secure Wired (Ethernet) Connections 2 Properties

Policy name: Secure Wired (Ethernet) Connections 2

Policy State

If enabled, NPS evaluates this policy while processing connection requests. If disabled, NPS does not evalue this policy.

☑ Policy enabled

Network connection method

Select the type of network access server that sends the connection request to NPS. You can select either the network access server type or Vendor specific, but neither is required. If your network access server is an 802.1X authenticating switch or wireless access point, select Unspecified.

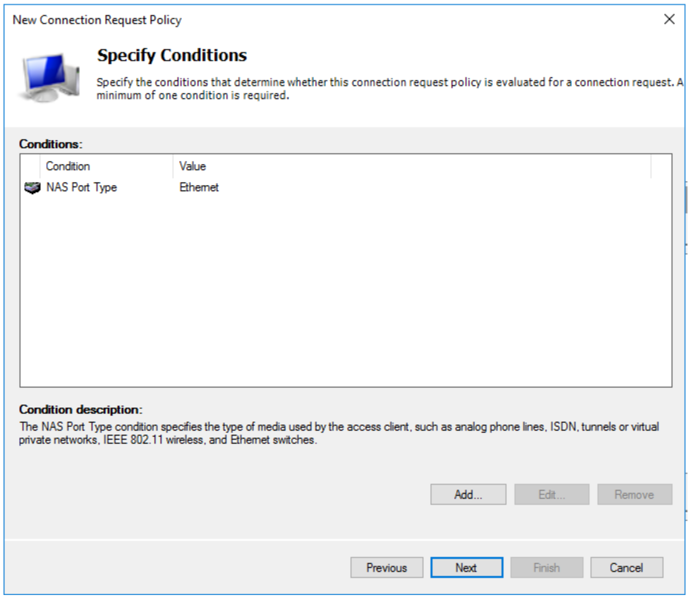
● Type of network access server:

Unspecified

○ Vendor specific:

10

- On the Specify Conditions page, click Add. Select NAS Port Type and click Add. Then check Ethernet under Common 802.1X connection tunnel types.



New Connection Request Policy

Specify Conditions

Specify the conditions that determine whether this connection request policy is evaluated for a connection request. A minimum of one condition is required.

Conditions:

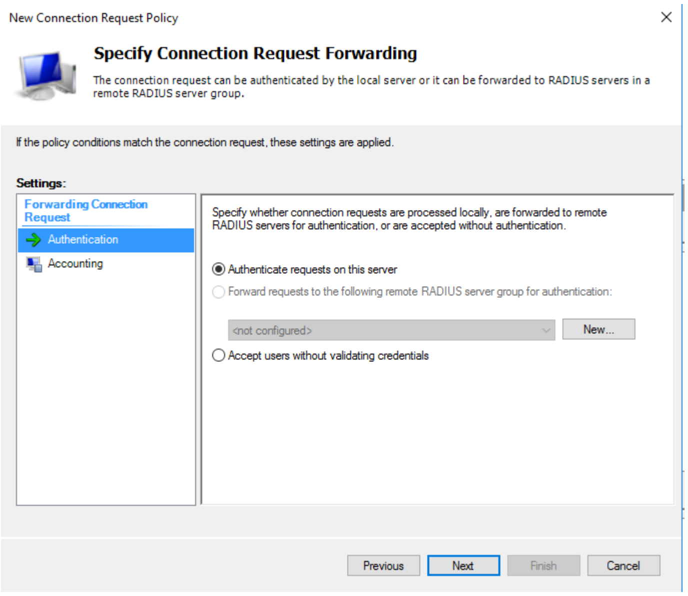
Condition	Value
NAS Port Type	Ethernet

Condition description:

The NAS Port Type condition specifies the type of media used by the access client, such as analog phone lines, ISDN, tunnels or virtual private networks, IEEE 802.11 wireless, and Ethernet switches.

- Click OK and click Next

- On the Specify Connection Request Forwarding page Click Next



New Connection Request Policy

Specify Connection Request Forwarding

The connection request can be authenticated by the local server or it can be forwarded to RADIUS servers in a remote RADIUS server group.

If the policy conditions match the connection request, these settings are applied.

Settings:

Specify whether connection requests are processed locally, are forwarded to remote RADIUS servers for authentication, or are accepted without authentication.

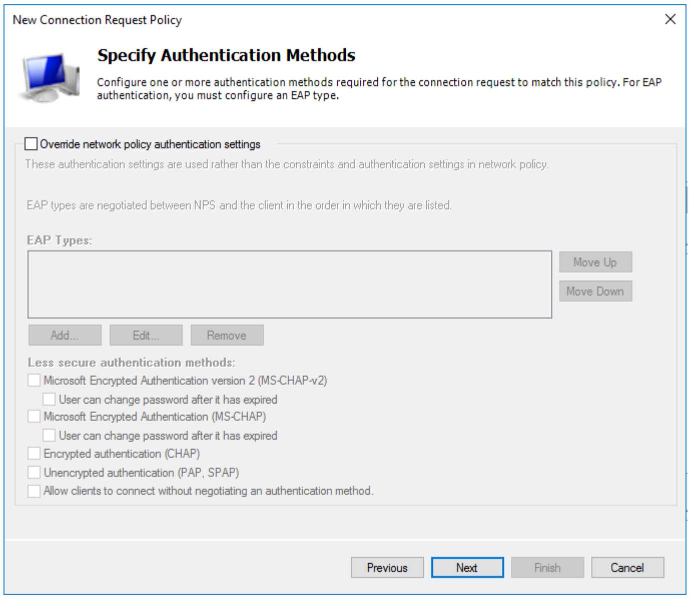
● Authenticate requests on this server

○ Forward requests to the following remote RADIUS server group for authentication:

<not configured>

○ Accept users without validating credentials

- On the Specify Authentication Methods page click Next



New Connection Request Policy

Specify Authentication Methods

Configure one or more authentication methods required for the connection request to match this policy. For EAP authentication, you must configure an EAP type.

□ Override network policy authentication settings

These authentication settings are used rather than the constraints and authentication settings in network policy.

EAP types are negotiated between NPS and the client in the order in which they are listed.

EAP Types:

Less secure authentication methods:

□ Microsoft Encrypted Authentication version 2 (MS-CHAP-v2)

□ User can change password after it has expired

□ Microsoft Encrypted Authentication (MS-CHAP)

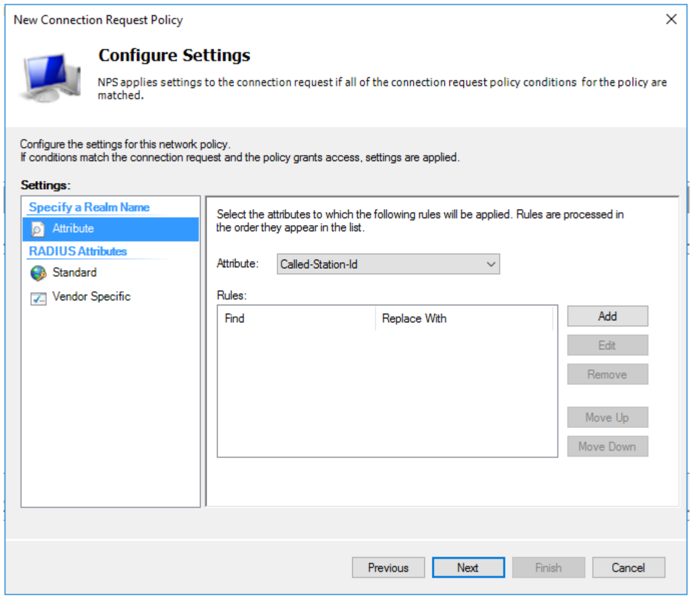
□ User can change password after it has expired

□ Encrypted authentication (CHAP)

□ Unencrypted authentication (PAP, SPAP)

□ Allow clients to connect without negotiating an authentication method.

- On the Configure Settings page click Next


New Connection Request Policy

Configure Settings

NPS applies settings to the connection request if all of the connection request policy conditions for the policy are matched.

Configure the settings for this network policy.
If conditions match the connection request and the policy grants access, settings are applied.

Settings:

Select the attributes to which the following rules will be applied. Rules are processed in the order they appear in the list.

Attribute: Called-Station-Id

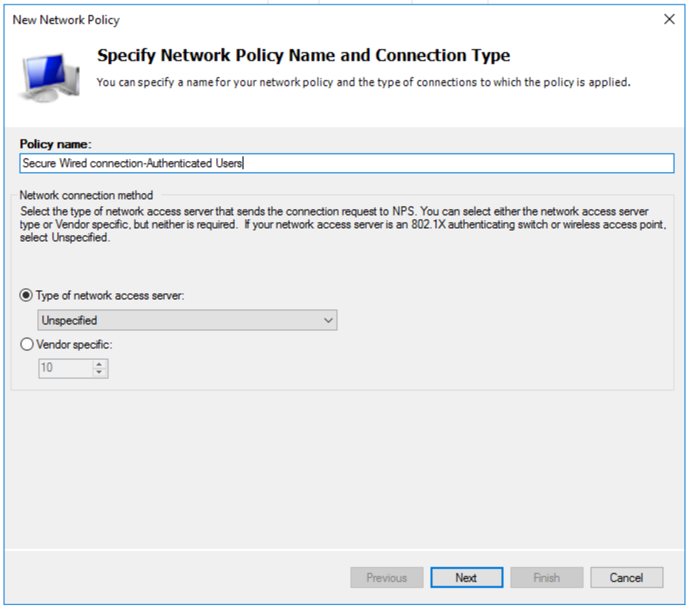
Rules:

Find	Replace With

- Click Finish

Configure Network Policy 802.1x Users

- From the Network Policy Server Console, right click on Network Policies and select New
- On the Specify Network Policy Name and Connection Type page, type a name for the policy and click Next



New Network Policy

Specify Network Policy Name and Connection Type

You can specify a name for your network policy and the type of connections to which the policy is applied.

Policy name:

Secure Wired connection-Authenticated Users

Network connection method

Select the type of network access server that sends the connection request to NPS. You can select either the network access server type or Vendor specific, but neither is required. If your network access server is an 802.1X authenticating switch or wireless access point, select Unspecified.

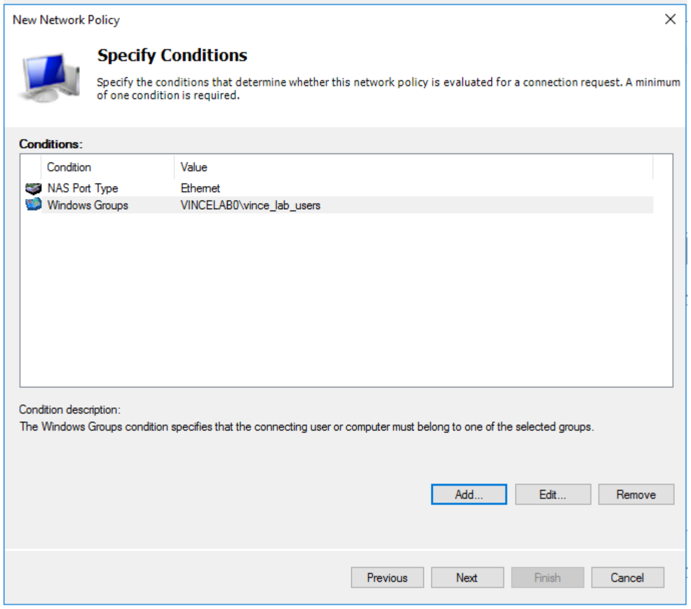
● Type of network access server:

Unspecified

○ Vendor specific:

10

- From the Select Conditions dialog box, select NAS Port Type (Ethernet) and click Add
- From the Select Condition dialog box, add the following Windows Groups: "What ever your users group is" and click Next



New Network Policy

Specify Conditions

Specify the conditions that determine whether this network policy is evaluated for a connection request. A minimum of one condition is required.

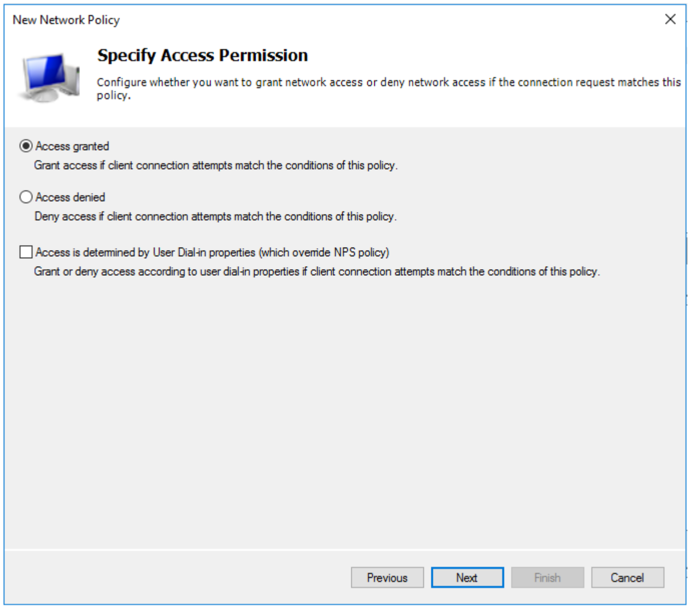
Conditions:

Condition	Value
NAS Port Type	Ethernet
Windows Groups	VINCELAB0\vince_lab_users

Condition description:

The Windows Groups condition specifies that the connecting user or computer must belong to one of the selected groups.

- On the Specify Access Permissions page, select Access Granted and click Next



New Network Policy

Specify Access Permission

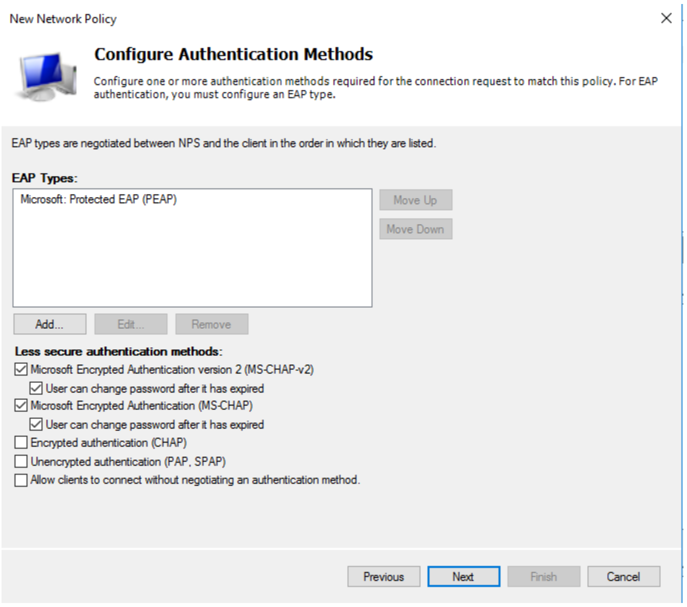
Configure whether you want to grant network access or deny network access if the connection request matches this policy.

● Access granted
Grant access if client connection attempts match the conditions of this policy.

○ Access denied
Deny access if client connection attempts match the conditions of this policy.

□ Access is determined by User Dial-in properties (which override NPS policy)
Grant or deny access according to user dial-in properties if client connection attempts match the conditions of this policy.

- On the Select EAP dialog box, select Microsoft: Protected EAP (PEAP)



New Network Policy

Configure Authentication Methods

Configure one or more authentication methods required for the connection request to match this policy. For EAP authentication, you must configure an EAP type.

EAP types are negotiated between NPS and the client in the order in which they are listed.

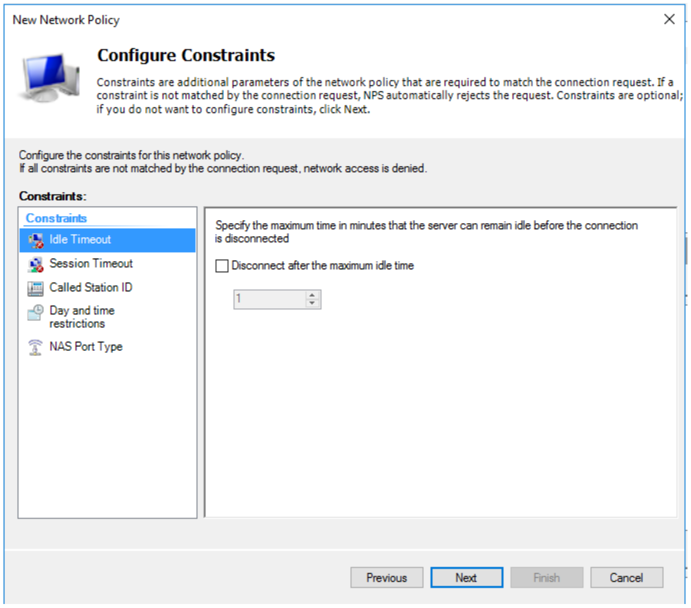
EAP Types:

Microsoft: Protected EAP (PEAP)

Less secure authentication methods:

☑ Microsoft Encrypted Authentication version 2 (MS-CHAP-v2)
☑ User can change password after it has expired
☑ Microsoft Encrypted Authentication (MS-CHAP)
☑ User can change password after it has expired
□ Encrypted authentication (CHAP)
□ Unencrypted authentication (PAP, SPAP)
□ Allow clients to connect without negotiating an authentication method.

- On the Configure Constraints page, click Next



New Network Policy

Configure Constraints

Constraints are additional parameters of the network policy that are required to match the connection request. If a constraint is not matched by the connection request, NPS automatically rejects the request. Constraints are optional; if you do not want to configure constraints, click Next.

Configure the constraints for this network policy.
If all constraints are not matched by the connection request, network access is denied.

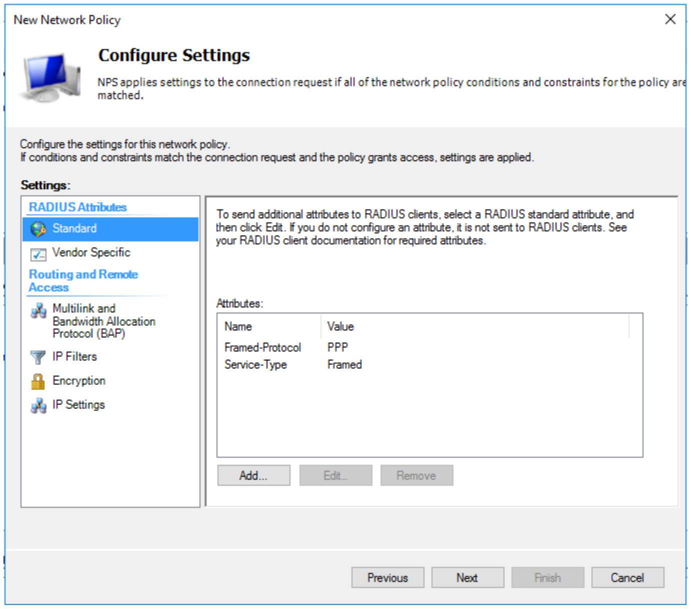
Constraints:

Specify the maximum time in minutes that the server can remain idle before the connection is disconnected

☐ Disconnect after the maximum idle time

1

- On the Configure Settings page click Next



New Network Policy

Configure Settings

NPS applies settings to the connection request if all of the network policy conditions and constraints for the policy are matched.

Configure the settings for this network policy.
If conditions and constraints match the connection request and the policy grants access, settings are applied.

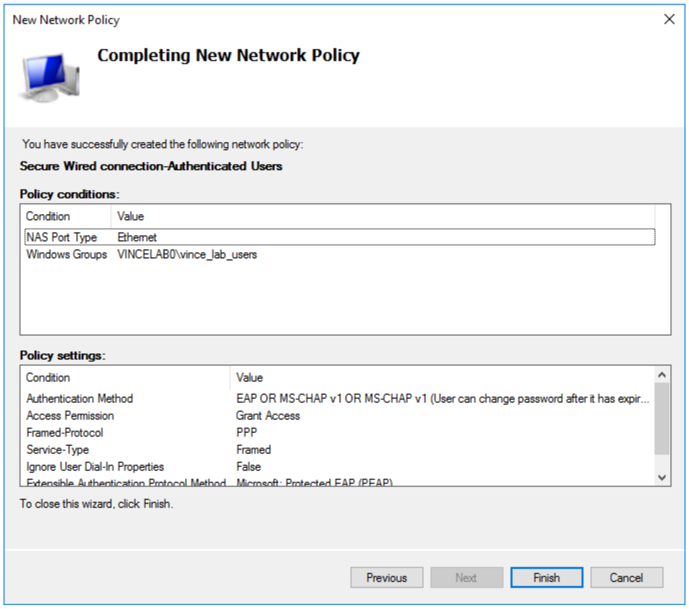
Settings:

To send additional attributes to RADIUS clients, select a RADIUS standard attribute, and then click Edit. If you do not configure an attribute, it is not sent to RADIUS clients. See your RADIUS client documentation for required attributes.

Attributes:

Name	Value
Framed-Protocol	PPP
Service-Type	Framed

- On the Completing New Network Policy page click Finish



New Network Policy

Completing New Network Policy

You have successfully created the following network policy:

Secure Wired connection-Authenticated Users

Policy conditions:

Condition	Value
NAS Port Type	Ethernet
Windows Groups	VINCELAB0\vince_lab_users

Policy settings:

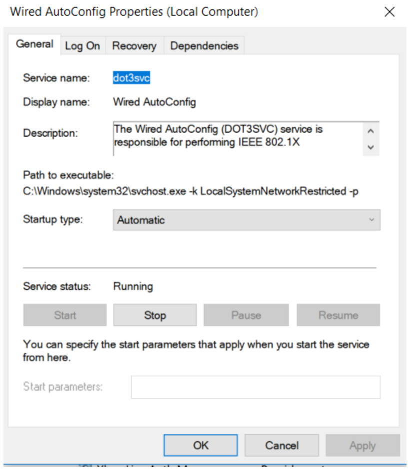
Condition	Value
Authentication Method	EAP OR MS-CHAP v1 OR MS-CHAP v1 (User can change password after it has expir...
Access Permission	Grant Access
Framed-Protocol	PPP
Service-Type	Framed
Ignore User Dial-In Properties	False
Extensible Authentication Protocol Method	Microsoft: Protected EAP (PEAP)

To close this wizard, click Finish.

A.3.1 Dell Workstations

Configuring Windows 10 client for 802.1x Authentication

- Navigate to Control Panel > Administrative > Services. Then from the list of services, right click on Wired AutoConfig and select Properties
- On the General tab, select Automatic under Startup type and click Start under Service Status



Wired AutoConfig Properties (Local Computer)

Service name: dot3svc

Display name: Wired AutoConfig

Description: The Wired AutoConfig (DOT3SVC) service is responsible for performing IEEE 802.1X

Path to executable:
C:\Windows\system32\svchost.exe -k LocalSystemNetworkRestricted -p

Startup type: Automatic

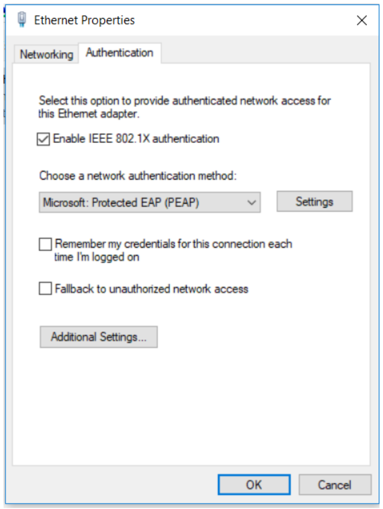
Service status: Running

You can specify the start parameters that apply when you start the service from here.

Start parameters:

- Click OK

- In the Control Panal open Network and sharing Center, and select Change adapter settings
- Right click on Local Area Connection and select Properties
- Select Authentication tab and select Enable IEEE 802.1X authentication
- Clear Remember my credentials for this connection each time I'm logged on and Fallback to unauthorized network access



Ethernet Properties

Networking | Authentication

Select this option to provide authenticated network access for this Ethernet adapter.

☑ Enable IEEE 802.1X authentication

Choose a network authentication method:

Microsoft: Protected EAP (PEAP) | Settings

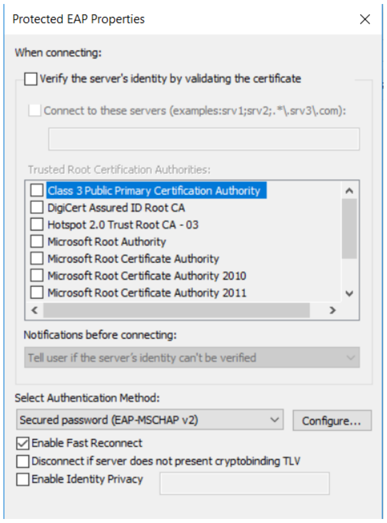
☐ Remember my credentials for this connection each time I'm logged on

☐ Fallback to unauthorized network access

Additional Settings...

OK | Cancel

- On the Choose a network authentication method list box, select Microsoft: Protected EAP (PEAP) and click Settings
- From the Select Authentication Method list box, select Secured password (EAP or MSCHAP v2) certificate and click OK



Protected EAP Properties

When connecting:

☐ Verify the server's identity by validating the certificate

☐ Connect to these servers (examples:srv1;srv2;.*\.srv3\.com):

Trusted Root Certification Authorities:

☐ Class 3 Public Primary Certification Authority
☐ DigiCert Assured ID Root CA
☐ Hotspot 2.0 Trust Root CA - 03
☐ Microsoft Root Authority
☐ Microsoft Root Certificate Authority
☐ Microsoft Root Certificate Authority 2010
☐ Microsoft Root Certificate Authority 2011

Notifications before connecting:

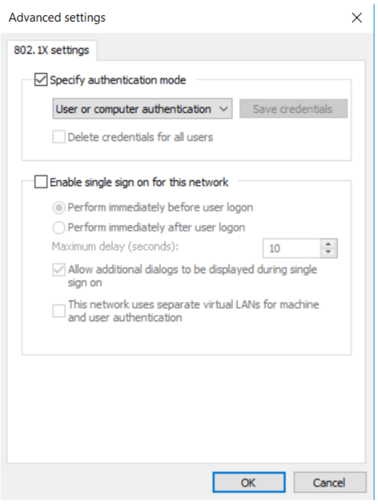
Tell user if the server's identity can't be verified

Select Authentication Method:

Secured password (EAP-MSCHAP v2) | Configure...

☑ Enable Fast Reconnect
☐ Disconnect if server does not present cryptobinding TLV
☐ Enable Identity Privacy

- Click Additional Settings, select Specify authentication mode and select User or Computer authentication from the list



Advanced settings

802.1X settings

☑ Specify authentication mode

User or computer authentication

□ Delete credentials for all users

□ Enable single sign on for this network

● Perform immediately before user logon

○ Perform immediately after user logon

Maximum delay (seconds): 10

☑ Allow additional dialogs to be displayed during single sign on

□ This network uses separate virtual LANs for machine and user authentication

- Click OK

A.4 SDN Installation and Setup

ONOS Installation

The ONOS install looks to install in the "/opt" directory on your workstation. These directions are locate on the ONOS wiki page [30], but we will walk through them as well:

- There shouldn't be a directory by that name by default, so the first thing we need to do is make the "/opt" directory and move to that directory.

```
sudo mkdir /opt
cd /opt
```

- Download ONOS from website

```
sudo wget -c http://downloads.onosproject.org/release/onos-1.14.0.tar.gz
```

- Extract the ONOS archive into the "/opt" directory

```
sudo tar xzf onos-1.14.0.tar.gz
```

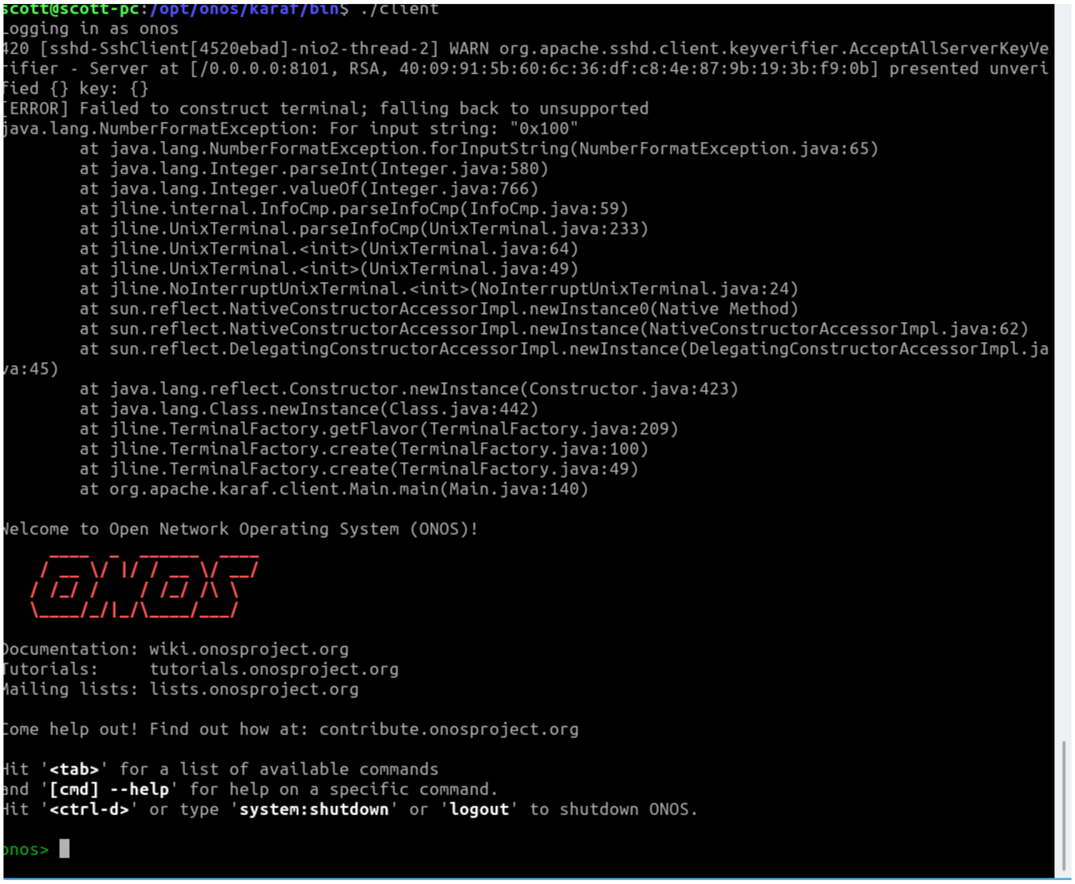
- Move the extracted directory to "onos" sudo mv onos-1.14.0 onos

How to Run ONOS

- To run or stop ONOS, you have to navigate to the "/opt/onos/bin" directory

```
./onos-service start or ./onos-service stop
```

- Once you start ONOS you will see the ONOS logo on your command line interface (CLI)

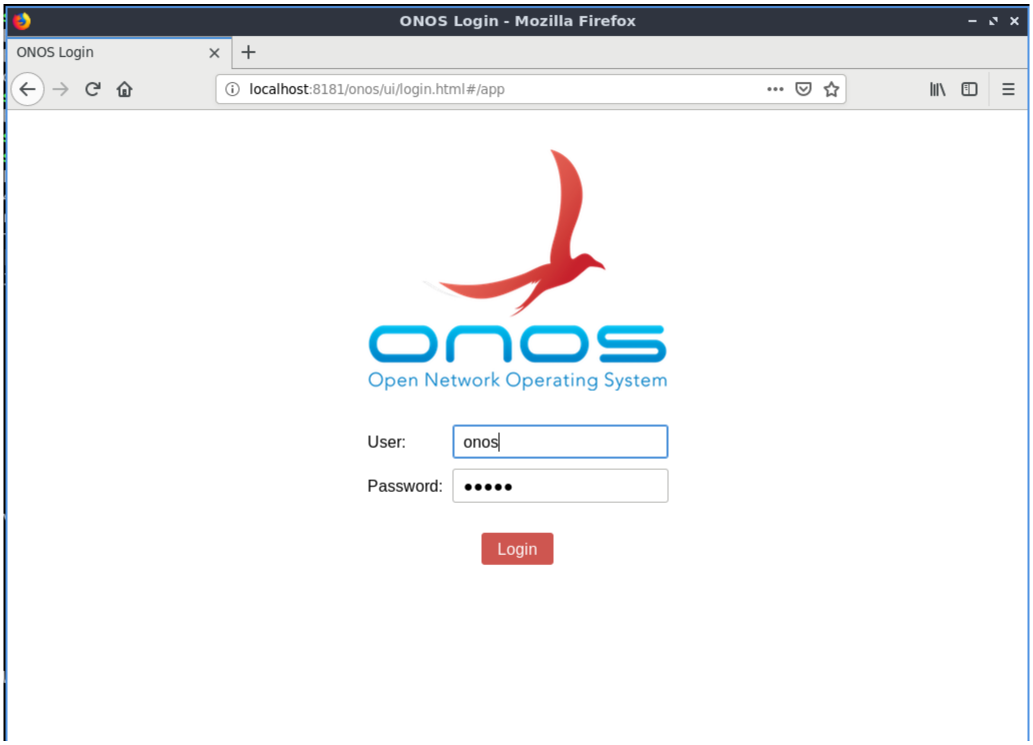


```
scott@scott-pc:/opt/onos/karaf/bin$ ./client
Logging in as onos
420 [sshd-SshClient[4520ebad]-nio2-thread-2] WARN org.apache.sshd.client.keyverifier.AcceptAllServerKeyVe
rifier - Server at [/0.0.0.0:8101, RSA, 40:09:91:5b:60:6c:36:df:c8:4e:87:9b:19:3b:f9:0b] presented unveri
fied {} key: {}
[ERROR] Failed to construct terminal; falling back to unsupported
java.lang.NumberFormatException: For input string: "0x100"
        at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
        at java.lang.Integer.parseInt(Integer.java:580)
        at java.lang.Integer.valueOf(Integer.java:766)
        at jline.internal.InfoCmp.parseInfoCmp(InfoCmp.java:59)
        at jline.UnixTerminal.parseInfoCmp(UnixTerminal.java:233)
        at jline.UnixTerminal.<init>(UnixTerminal.java:64)
        at jline.UnixTerminal.<init>(UnixTerminal.java:49)
        at jline.NoInterruptUnixTerminal.<init>(NoInterruptUnixTerminal.java:24)
        at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
        at sun.reflect.NativeConstructorAccessorImpl.newInstance(NativeConstructorAccessorImpl.java:62)
        at sun.reflect.DelegatingConstructorAccessorImpl.newInstance(DelegatingConstructorAccessorImpl.ja
va:45)
        at java.lang.reflect.Constructor.newInstance(Constructor.java:423)
        at java.lang.Class.newInstance(Class.java:442)
        at jline.TerminalFactory.getFlavor(TerminalFactory.java:209)
        at jline.TerminalFactory.create(TerminalFactory.java:100)
        at jline.TerminalFactory.create(TerminalFactory.java:49)
        at org.apache.karaf.client.Main.main(Main.java:140)

Welcome to Open Network Operating System (ONOS)!
     ____  _  ______  ____
    / __ \/ |/ / __ \/ __/
   / /_/ /    / /_/ /\ \
   \____/_/|_/\____/___/

Documentation: wiki.onosproject.org
Tutorials:     tutorials.onosproject.org
Mailing lists: lists.onosproject.org

Come help out! Find out how at: contribute.onosproject.org

Hit '<tab>' for a list of available commands
and '[cmd] --help' for help on a specific command.
Hit '<ctrl-d>' or type 'system:shutdown' or 'logout' to shutdown ONOS.

onos> 
```

- You can also view ONOS in a web browser. Type the following into your web browser and press ENTER

```
localhost:8181/onos/UI/login.html#/app
```

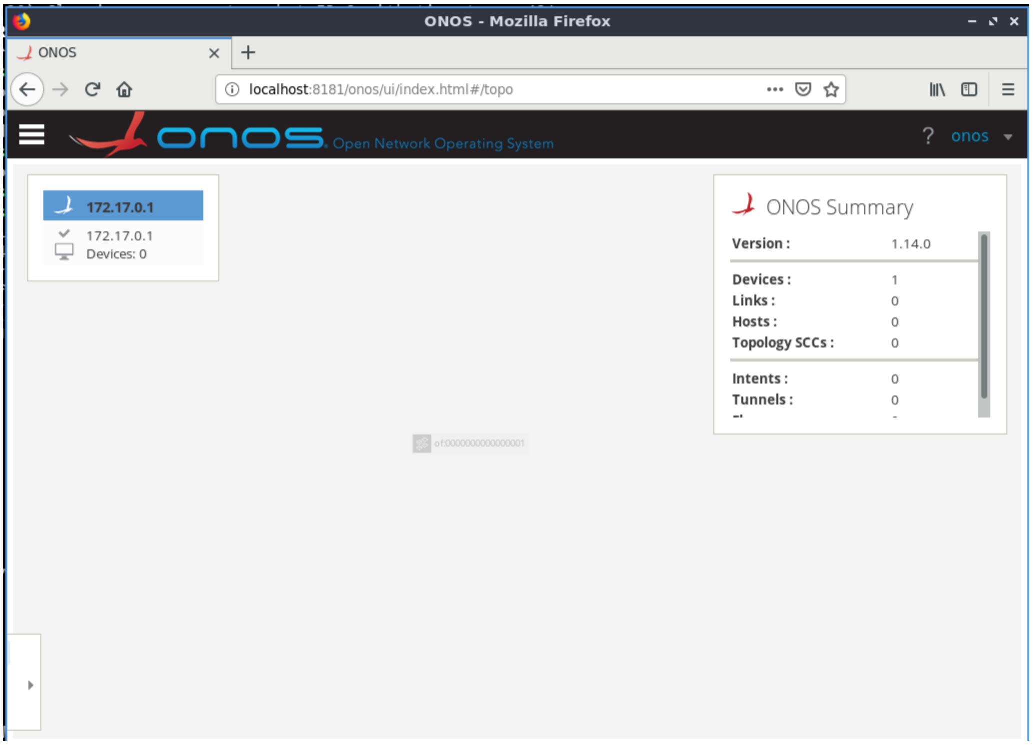
- The following will show up in the browser. The username is "onos" and the password is "rocks"


ONOS Login - Mozilla Firefox
ONOS Login
localhost:8181/onos/ui/login.html#/app
onos
Open Network Operating System
Login

User: onos

Password: •••••

- Once you log in, you will get a graphic user interface (GUI) that displays your current network topology. It should be empty for the moment


ONOS - Mozilla Firefox
ONOS
localhost:8181/onos/ui/index.html#/topo
onos Open Network Operating System
?
onos
of:0000000000000001

172.17.0.1

✓ 172.17.0.1
Devices: 0

ONOS Summary

Version :	1.14.0
Devices :	1
Links :	0
Hosts :	0
Topology SCCs :	0
Intents :	0
Tunnels :	0

ONOS Applications

One of SDN's advantages over legacy networks are their programmability. With ONOS, you get a considerable amount of applications built in from previous developers so you don't have to recreate some of the most general use cases, such as setting up flows for hosts. One application that we want to use that was not installed was the AAA application. It was built on CORD and since CORD is built on top of ONOS technology, we can add it into our ONOS build.

- Once you download the AAA application snapshot from github, you will use the following command to upload the application into ONOS

```
onos-app 172.17.0.1<-"this is ONOS IP address by default" install <path to "aaa-1.2-SNAPSHOT.oar">
```

- Then we need to supply the AAA application with RADIUS configuration, so we create a aaa-conf.json file to provide the information. Because RADIUS will be setup on the same workstation the IP address will be 127.0.0.1 for local host.

```
        {
```

```
        "apps": {
            "org.opencord.aaa" : {
                "AAA" : {
                    "radiusIp": "127.0.0.1",
                    "radiusServerPort": "1812",
                    "radiusSecret": "password"
                }
            }
        }
    }
```

- Then we upload the aaa-conf.json file to ONOS

```
onos-netcfg 172.17.0.1 aaa-conf.json
```

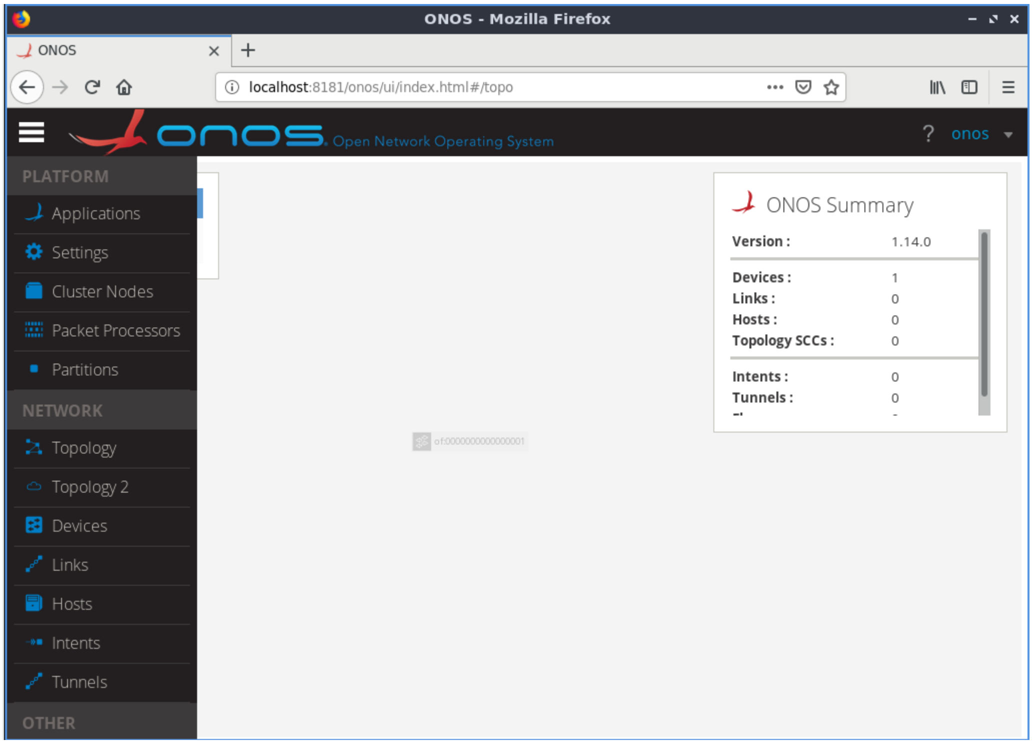
- You can view the apps available on your ONOS controller by running the following command from CLI

```
apps -a
```

- You can also run this command to see what apps you have running

```
apps -a -s
```

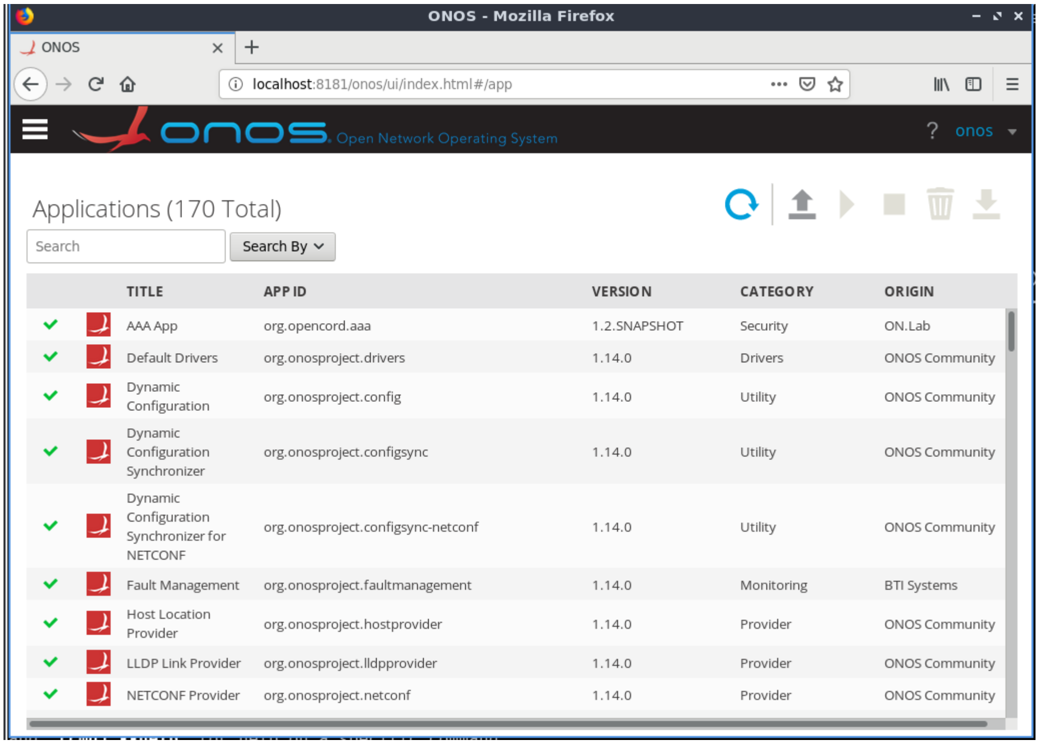
- Or you can view all your apps, active or not active, from the web browser by clicking on the menu button in the top left corner of your browser



ONOS Summary

Version :	1.14.0
Devices :	1
Links :	0
Hosts :	0
Topology SCCs :	0
Intents :	0
Tunnels :	0

- Then click on "Applications" from the menu


ONOS - Mozilla Firefox
ONOS
localhost:8181/onos/ui/index.html#/app
onos Open Network Operating System
onos
Search
Search By

Applications (170 Total)

		TITLE	APP ID	VERSION	CATEGORY	ORIGIN
✓		AAA App	org.opencord.aaa	1.2.SNAPSHOT	Security	ON.Lab
✓		Default Drivers	org.onosproject.drivers	1.14.0	Drivers	ONOS Community
✓		Dynamic Configuration	org.onosproject.config	1.14.0	Utility	ONOS Community
✓		Dynamic Configuration Synchronizer	org.onosproject.configsync	1.14.0	Utility	ONOS Community
✓		Dynamic Configuration Synchronizer for NETCONF	org.onosproject.configsync-netconf	1.14.0	Utility	ONOS Community
✓		Fault Management	org.onosproject.faultmanagement	1.14.0	Monitoring	BTI Systems
✓		Host Location Provider	org.onosproject.hostprovider	1.14.0	Provider	ONOS Community
✓		LLDP Link Provider	org.onosproject.lldpprovider	1.14.0	Provider	ONOS Community
✓		NETCONF Provider	org.onosproject.netconf	1.14.0	Provider	ONOS Community

A.4.1 Mininet

Mininet will be used to simulate our hosts and switches for our SDN implementation. It allows for flexible and quick topology building and testing.

Installation

Run the following command to install Mininet

```
sudo apt-get install mininet
```

Topology Setup

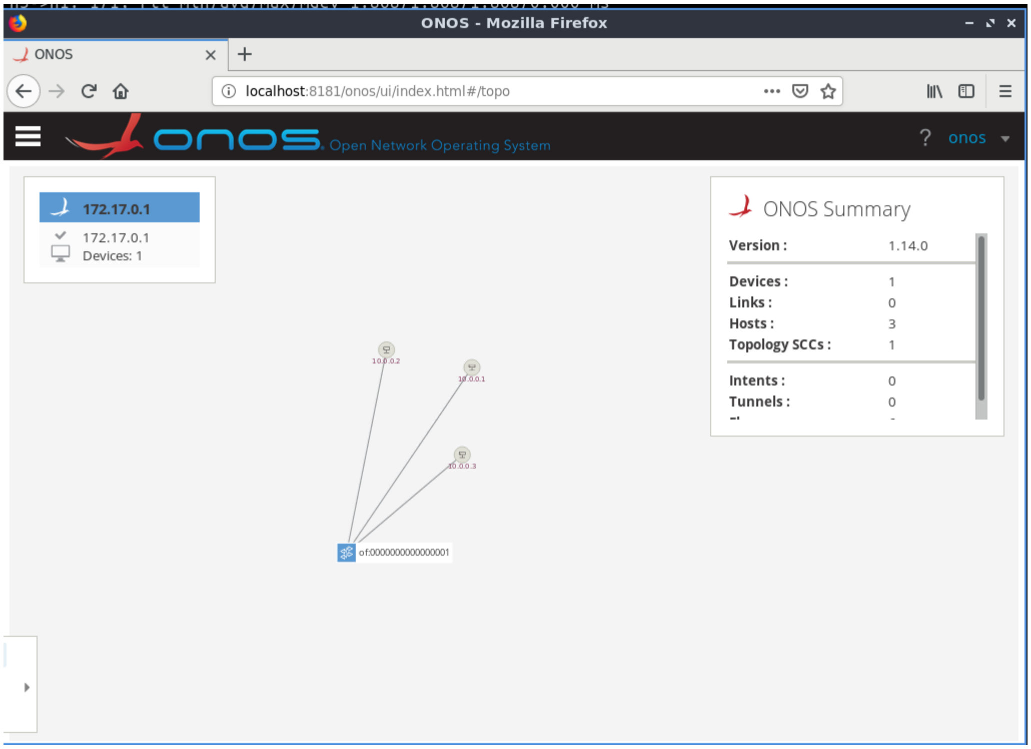
A network topology refers to the layout or arrangement of network communication devices (nodes, links, etc.) [41]. We are creating a simple topology that will include one open vswitch and three hosts. They will connect to the ONOS controller remotely on port 6633.

```
sudo mn --topo=tree,1,3 --controller=remote,ip=172.17.0.1:6633 --mac
```

Once you build this topology, you will need to ping all hosts to create flows.

```
pingallfull
```

You will then be able to view the topology in the web browser on ONOS.


ONOS - Mozilla Firefox
ONOS
localhost:8181/onos/ui/index.html#/topo
onos Open Network Operating System
onos
10.0.0.2
10.0.0.1
10.0.0.3
of:0000000000000001

172.17.0.1
172.17.0.1
Devices: 1

ONOS Summary

```
Version :          1.14.0
Devices :          1
Links :            0
Hosts :            3
Topology SCCs :    1
Intents :          0
Tunnels :          0
```

A.4.2 FreeRADIUS

For FreeRADIUS, we have to install the software, configure EAP as our authentication method, create certificates with the built in CA and create a user account.

Installation

Run the following command to install FreeRADIUS

```
sudo apt-get install freeradius
```

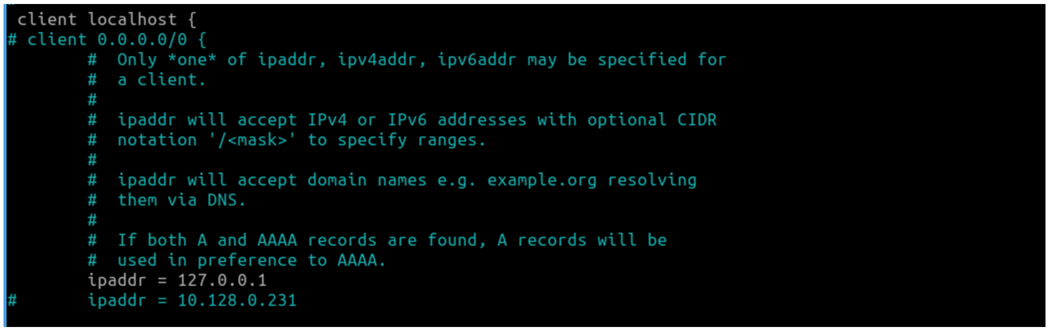
Configuration

We are going to walk through the configuration files that need to be altered for us to use EAP authentication for FreeRADIUS.

- navigate to etc/freeradius/3.0/clients.conf file to add a RADIUS client.

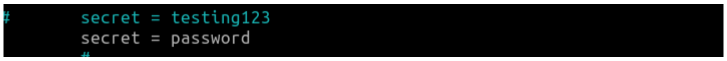
```
cd /etc/freeradius/3.0/
nano clients.conf
```

- Since the workstation is the client, the client will be localhost and the IP address will be 127.0.0.1



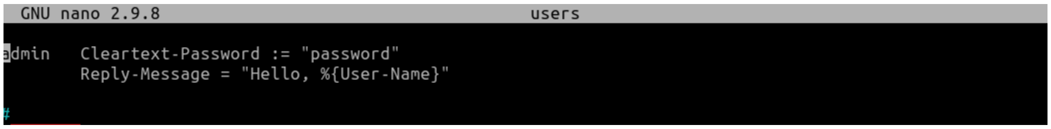
```
 client localhost {
# client 0.0.0.0/0 {
        #  Only *one* of ipaddr, ipv4addr, ipv6addr may be specified for
        #  a client.
        #
        #  ipaddr will accept IPv4 or IPv6 addresses with optional CIDR
        #  notation '/<mask>' to specify ranges.
        #
        #  ipaddr will accept domain names e.g. example.org resolving
        #  them via DNS.
        #
        #  If both A and AAAA records are found, A records will be
        #  used in preference to AAAA.
        ipaddr = 127.0.0.1
#       ipaddr = 10.128.0.231
```

- Make sure you change the shared secret password. Since this is a closed test bed, we will keep simple for easy configuration purposes.



```
#       secret = testing123
        secret = password
```

- Next, navigate to the users file to add a user
- Add a user and password

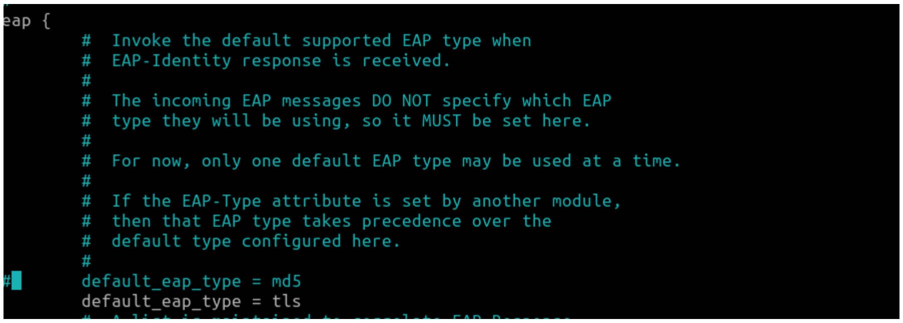


```
 GNU nano 2.9.8                                    users

admin   Cleartext-Password := "password"
        Reply-Message = "Hello, %{User-Name}"

#
```

- Navigate to EAP file to configure settings

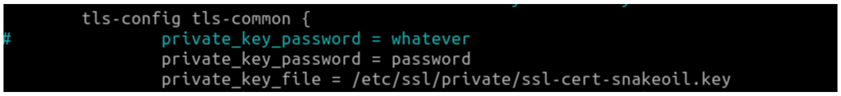
```
cd /etc/freeradius/3.0/mods-available
nano eap
```

- In the EAP file, change the default EAP type from MD5 to TLS



```
eap {
        #  Invoke the default supported EAP type when
        #  EAP-Identity response is received.
        #
        #  The incoming EAP messages DO NOT specify which EAP
        #  type they will be using, so it MUST be set here.
        #
        #  For now, only one default EAP type may be used at a time.
        #
        #  If the EAP-Type attribute is set by another module,
        #  then that EAP type takes precedence over the
        #  default type configured here.
        #
#       default_eap_type = md5
        default_eap_type = tls
```

- Under "tls-config tls-common" section, change the "private key password".



```
        tls-config tls-common {
#               private_key_password = whatever
                private_key_password = password
                private_key_file = /etc/ssl/private/ssl-cert-snakeoil.key
```

- Navigate to the etc/freeradius/3.0/certs directory
- In the certs directory, there are three files that need to updated; server.cnf file, ca.cnf and client.cnf. Change the "input password" and "output password" in all of those files.

How to Run FreeRADIUS

To run RADIUS you will use the following command

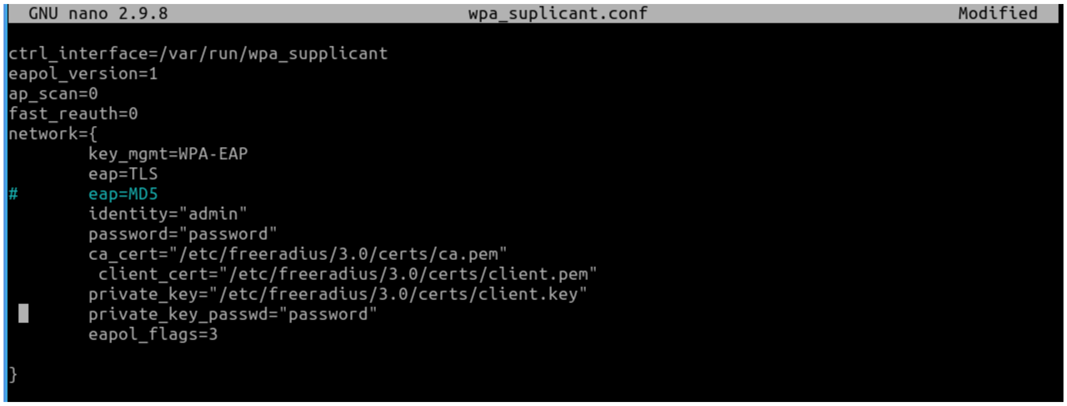
```
sudo freeradius -X
```

When you run this command the first time, it will create your certificates for the server, client and CA. You will need to move a copy of the ca.pem, client.pem and client.key to another folder for users.

```
cp ca.pem /etc/certs
cp clients.pem /etc/certs
cp client.key /etc/certs
```

A.4.3 WPA-SUPPLICANT

WPA-Supplicant is an implementation of the WPA supplicant that runs on your host workstation. It allows you simulate WPA key negotiation with a WPA authenticator and EAP authentication with an authentication server. To use the WPA-supplicant program, you have to create a wpa-supplicant configuration file with the authentication method, username and password, and certificates depending on the authentication method being used. Since we are using EAP-TLS, this is the configuration that was created.



```
  GNU nano 2.9.8                     wpa_suplicant.conf                      Modified

ctrl_interface=/var/run/wpa_supplicant
eapol_version=1
ap_scan=0
fast_reauth=0
network={
        key_mgmt=WPA-EAP
        eap=TLS
#       eap=MD5
        identity="admin"
        password="password"
        ca_cert="/etc/freeradius/3.0/certs/ca.pem"
         client_cert="/etc/freeradius/3.0/certs/client.pem"
        private_key="/etc/freeradius/3.0/certs/client.key"
        private_key_passwd="password"
        eapol_flags=3

}
```

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] Brocade ICX 7750 Switch. Brocade. [Online]. Available: http://www.dataswitchworks.com/datasheets/brocade-icx-7750-ds.pdf

[2] Brocade MLXe enterprise switches. (2019). [Online]. Available: https://www.dataswitchworks.com/datasheets/Switches/brocade-mlxe-enterprise-switches-ds.pdf

[3] PA-7050. Palo Alto. [Online]. Available: http://www.cc.com.pl/pl/prods/paloaltonetworks/pdf/pa-7050.pdf

[4] Tutorial: network access control (NAC). (July 17, 2007). *Network Computing*. [Online]. Available: https://www.networkcomputing.com/careers/tutorial-network-access-control-nac/880346581

[5] Network access control. (n.d.). *Wikipedia*. [Online]. Available: https://en.wikipedia.org/wiki/Network_Access_Control. Accessed March 1, 2018.

[6] Drew Robb. 9 Top network access control (NAC) solutions. [Online]. Available: https://www.esecurityplanet.com/products/top-network-access-control-solutions.html#impulse. Posted July 7, 2017.

[7] G. G. X. Yu-Wei Eric Sung, Sanjay G. Rao and D. A. Maltz, "Towards systematic design of enterprise networks," in *IEEE/ACM Transactions on Networking, 2008*, Madrid, Spain, 2011, vol. 19, pp. 695–708.

[8] Understanding and configuring VLANs. (Updated Feb. 15, 2018). Cisco. [Online]. Available: https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/vlans.html

[9] M. Meyers, *CompTIA Network+ Certification All-in-One Exam Guide, Seventh Edition (Exam N10-007)*, 7th ed. New York: McGraw-Hill, 2018.

[10] E. A. Arthur Conklin, *Principles of Computer Security: CompTIA Security+ and Beyond, Fifth Edition*, 5th ed. New York: McGraw-Hill, 2018.

[11] Joel Snyder. (2010). What is 802.1X? Everything you need to know about the Wi-Fi standard. *NETWORK WORLD*. [Online]. Available: https://www.networkworld.com/article/2216499/security/wireless-what-is-802-1x.html

[12] Microsoft. (2018, May 30). RADIUS authentication, authorization, and accounting. [Online]. Available: https://docs.microsoft.com/en-us/windows/desktop/nps/ias-radius-authentication-and-accounting

[13] P. C. et al. IEEE 802.1X remote authentication dial in user service (RADIUS) usage guidelines. [Online]. Available: https://tools.ietf.org/html/rfc3580

[14] TargetTech.com. (LDAP) Lightweight directory access protocol. Search Mobile Computing. [Online]. Available: https://searchmobilecomputing.techtarget.com/definition/LDAP. Updated November 2008.

[15] Hack2Secure. How LDAP protocol works. [Online]. Available: https://www.hack2secure.com/blogs/how-ldap-protocol-works. Accessed March 26, 2018.

[16] E. J. Sermersheim. (2006, June). Lightweight directory access protocol (LDAP): The protocol. RFC 4511. Available: https://www.ietf.org/rfc/rfc4511.txt

[17] P. Goransson, C. Black, and T. Culver, *Software Defined Networks, Second Edition: A Comprehensive Approach*, 2nd ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2016.

[18] E. A. Anders Nguyen. (2015, Mar.). OpenFlow switch specification. [Online]. Available: https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf

[19] CAM (content addressable memory) vs TCAM (ternary content addressable memory). (Updated Aug 23, 2017). Cisco. [Online]. Available: https://community.cisco.com/t5/network-architecture-documents/cam-content-addressable-memory-vs-tcam-ternary-content/ta-p/3107938

[20] 10 SDN platform options. (March 9, 2016). *Network Computing*. [Online]. Available: https://www.networkcomputing.com/networking/10-sdn-platform-options/1101283517

[21] J. Loos. (2012). Implementing IEEE 802.1x for wired networks. SANS Institute. SANS Institute InfoSec Reading Room.

[22] K. Benzekki, A. E. Fergougui, and A. E. A. E. Belrhiti, “Devolving IEEE 802.1x authentication capability to data plane in software-defined networking (SDN) architecture,” *Security and Communication Networks*, vol. 9, no. 17, pp. 4369–4377, 2016.

[23] J. Matias, J. Garay, A. Mendiola, N. Toledo, and E. Jacob, “FlowNAC: Flow-based network access control,” in *Third European Workshop on Software Defined Networks*, 2014, p. 6.

[24] D. M. F. Mattos and O. C. M. B. Duarte, “Authflow: Authentication and access control mechanism for software defined networking,” *Annales des Télécommunications*, vol. 71, no. 11-12, pp. 607–615, 2016.

[25] Y. Yamasaki, Y. Miyamoto, J. Yamato, H. Goto, and H. Sone, “Flexible access management system for campus VLAN based on OpenFlow,” in *Applications and the Internet (SAINT), 2011 IEEE/IPSJ 11th International Symposium on*, 2011, pp. 347–351.

[26] S. Kinoshita, T. Watanabe, J. Yamato, H. Goto, and H. Sone, “Implementation and evaluation of an openflow-based access control system for wireless LAN roaming,” in *36th Annual IEEE Computer Software and Applications Conference Workshops, COMPSAC 2012, Izmir, Turkey, July 16-20, 2012*, 2012, pp. 82–87. Available: https://doi.org/10.1109/COMPSACW.2012.25

[27] Private network information for thesis - JIRA. (summer 2018). ITACS, Naval Postgraduate School. Monterey, CA. Notes from NOC ticket with ITACS network operations technician Mike Williams.

[28] Network robustness. (2018). Complexity Labs. [Online]. Available: https://complexitylabs.io/network-robustness-resilience/

[29] Techopedia. (2018). Scalability. [Online]. Available: https://www.techopedia.com/definition/9269/scalability

[30] ONOS. (2017, Sep.). *Wiki*. [Online]. Available: https://wiki.onosproject.org/display/ONOS/WikiHome

[31] CORD (Central Office Re-architected as a Datacenter): the killer app for SDN and NFV. *Open CORD*. [Online]. Available: https://opencord.org/. Accessed 2018.

[32] DaMule. (2013, Nov). Brocade ICX 6450-48P 48-port gigabit ethernet switch - ICX6450-48P - fixed (managed) switches. *CDW.com*. [Online]. Available: https://www.cdw.com/product/brocade-icx-6450-48p-48-port-gigabit-ethernet-switch/2584728

[33] *Dell Search*. [Online]. Available: https://www.dell.com/koa/search?q=sdn#q=sdn&t=default&sort=relevancy&layout=card&@dpsalessegment:radioGroup=bsd. Accessed 2018.

[34] Five SDN benefits enterprises should consider. (2013). [Online]. Available: https://www.networkcomputing.com/networking/five-sdn-benefits-enterprises-should-consider/70381323

[35] 7 Advantages of software-defined networking. (2017). [Online]. Available: https://imaginenext.ingrammicro.com/trends/august-2017/7-advantages-of-software-defined-networking

[36] Configuring interVLAN routing and ISL/802.1Q trunking on a Catalyst 2900XL/3500XL/2950 switch using an external router. (Updated November 22, 2005). Cisco. [Online]. Available: https://www.cisco.com/c/en/us/support/docs/lan-switching/inter-vlan-routing/14976-50.html

[37] Brocade ICX7750-48F ICX 7750 with 48 10GBE SFP and 6x 40GBE QSFP switch. (2019). [Online]. Available: https://www.ebay.com/p/Brocade-ICX7750-48F-ICX-7750-with-48-10GBE-SFP-and-6x-40GBE-QSFP-Switch/4008480878?iid=332933171042&chn=ps

[38] "Brocade icx 7750-48f layer 3 switch," 2019. Available: https://www.newegg.com/Product/Product.aspx?Item=9SIA25V6JE7364&ignorebbr=1&nm_mc=KNC-GoogleMKP-PC&cm_mmc=KNC-GoogleMKP-PC-_-pla-IPC+Store-_-Network+-+Switches-_-9SIA25V6JE7364&gclid=CjwKCAiAyMHhBRBIEiwAkGN6fEBVnUdaaB36mGrQuTGoi4W6sPOKDSYVgxd-3nriGn0bCvox7FVIfhoC1YEQAvD_BwE&gclsrc=aw.ds

[39] Brocade ICX 7450-48 48-port 10/100/1000 managed ethernet switch. (2019). [Online]. Available: https://www.pcliquidations.com/p82303-brocade-icx-7450-48?utm_source=google&utm_medium=cse&utm_term=82303&r160164167166161&gclid=CjwKCAiAyMHhBRBIEiwAkGN6fFqdTf33J7yhb1EA9zesL1aGnXwkTr2uoJ6eI3LI8wwWNKpDIIMHMhoCEp0QAvD_ BwE

[40] Brocade ICX 7450-48F - Switch - L3 - managed - 48 x Gigabit SFP - rack-mountable. (2019). [Online]. Available: http://www.nextwarehouse.com/item/?2813365_g10e&gclid=CjwKCAiAyMHhBRBIEiwAkGN6fGS0yd_bEQ5K-FCklmBg3YIwm7MDQylXZVDKTdancy-s3AQ2Xg3XvRoC7sYQAvD_BwE

[41] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. (2004, Feb.). Measuring ISP topologies with rocketfuel. *IEEE/ACM Trans. Netw.* Piscataway, NJ, USA. [Online]. *12*(1). pp. 2–16. Available: http://dx.doi.org/10.1109/TNET.2003.822655

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California